Underwater Acoustic Imaging: A Simulation Program and Related Theory

David G. Blair and Stuart D. Anstee

DSTO-TN-0274

20001205 051

Underwater Acoustic Imaging: A Simulation Program and Related Theory

David G. Blair and Stuart D. Anstee

Maritime Operations Division
Aeronautical and Maritime Research Laboratory

DSTO-TN-0274

ABSTRACT

To aid in the design of a high-resolution sonar (range and transverse resolution approaching 1 mm), a program, POINTSPR, was produced. POINTSPR predicts the image obtained from an active acoustic imaging system with a spherical projector, the target being modelled by a small number of point objects. The waveforms available include chirps and short tonebursts. The receiving elements available include a point and a square piston. A wide variety of receiving array geometries are available, through features including the random placement of elements and the placing of multiple copies of a 'tile' of elements. Monte Carlo calculations can be run automatically. Sample graphs, which are 1-D or 2-D slices through the 3-dimensional (3-D) image, are given.

The theory of image-forming, dechirping and the quadrature part for a real sonar is presented, including 'delay-and-add' image-forming, known to apply even in the near-field, broadband, 3-D-imaging situation. The report also gives many formulae for image intensity within the simulation model. For a chirp of finite length, certain approximations, often applied, are studied numerically. Formulae, possibly new, are obtained for the response of square elements to a chirp.

RELEASE LIMITATION

Approved for public release

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

Underwater Acoustic Imaging: A Simulation Program and Related Theory

Executive Summary

To counter the threat of sea mines, Australian defence policy recognises the need for an enhanced capability of identifying objects that have been classified as minelike. Optical methods of identification—the use of a video camera—fail when the medium is very turbid. For this reason DSTO is working with industry to develop a high-resolution acoustic imaging system. This sonar is intended to 'see' detail of size approaching 1 mm per metre of range.

The present report has three aims. The first is to describe a simulation program, POINTSPR, which can be used as a tool in designing array systems of this type. POINTSPR predicts the image obtained from an active underwater system with a spherical projector. The predictions are subject to a number of assumptions; in particular, the program models the target by a small number of point objects. The waveforms available in the simulation are of two types. The first is a short toneburst (a toneburst being obtained from a sine wave by applying two sharp cut-offs). The second waveform is a chirp, or burst with a linearly varying frequency; in this case a process of 'dechirping' is always applied to the received signal.

In the simulation, each sensor element of the receiving array is either a point transducer or a square piston. A wide variety of arrays can be generated, through features that include the random placement of elements and the placement of multiple copies of a preconfigured square 'tile' of elements. The program outputs three 1-D or 2-D slices through the 3-dimensional (3-D) image; sample graphs are included in the report. Quantisation error, due to recording each voltage to a limited number of binary digits, is modelled under certain conditions.

The second aim of the report is to present algebraic formulae and numerical trends that hold within the simulation model.

The third aim is to give the theory of image-forming, dechirping and the quadrature part (demodulation). While these items of theory are not new, their presentation in the context of sonar is believed to fill a gap. The image-forming method, which uses the 'delay-and-add' procedure, applies even in the near-field, broadband, three-dimensional-image situation that mine imaging faces.

Contents

1. Introduction

1.1 General

In countering sea mines, Australian defence policy recognises the need for an enhanced capability of identifying objects once they have been detected and classified as minelike. This 'identification' phase currently uses a video camera mounted on an underwater remotely operated vehicle (ROV). But under turbid conditions—often encountered—optical methods fail. A promising alternative would be the development of a high-resolution acoustic imaging system. Such a system would also have commercial applications outside the defence area, for example in inspecting oil drilling rigs for cracks. The system could be ROV-based but could, for example, be diver-held.

Possible acoustic systems of this type have been discussed by Truver [1995] and Verveur [1990], and in some papers in Bottoms *et al.* [1995]. Cuschieri and Cao [1992] have presented the design concepts for such a system, while Knudsen [1989] has described a prototype imaging system actually built, albeit with lower resolution. Bahl and Powers [1990] attempted a realistic computer simulation of the imaging process of a sector-scan sonar.

The acoustic mine imaging (AMI) program was developed from 1991 onwards by staff of the Maritime Operations Division (MOD) of DSTO [Jones 1996]. Its aim is the development of such an imaging system with both a lateral resolution and a range resolution of 1 mm at 1 m range. Throughout most of the program, the bulk of the work has been contracted out to industry (Thomson Marconi Sonar), with MOD monitoring the technical risk and making a contribution to the research. Two successful trials were conducted at Darwin by Thomson Marconi Sonar, CSIRO and MOD. In the first trial in 1995, the innovative acoustic sensor technology was demonstrated. The second trial in 1998 demonstrated the acquisition of images in seawater of medium turbidity.

Of the studies carried out within MOD, the early work—discussed by Jones [1994]—included an examination of the likely disruptive effect of the scattering of acoustic waves by the medium [Thuraisingham 1994a, 1994b], studies on the feasibility of sparse array technology [Blair *et al.* 1994] and a literature survey on array design [Blair 1993]. More recently, problems in achieving the rapid image-forming required have been addressed [Blair and Jones 1998, Blair 1997]. (By 'image-forming' we mean beamforming in three dimensions upon receive.)

The array in the proposed system is two-dimensional (2-D) and the resulting image is three-dimensional. It is desirable to be able to predict the properties of the array system before building it. The present report contributes in this area by reporting a simulation program and by presenting some of the theory of the real and the simulated system.

The report has three aims. The first is to describe a simulation program, POINTSPR,[1] that calculates and displays slices through the image. The discussion includes the simulation model on which POINTSPR is based, the resulting formulae, the derivations and the algorithms used.

The second aim is to present results concerning the image intensity obtained within the simulation model. Many of these results are analytical; others are obtained by analysing numerical outputs. In regard to graphs of numerical outputs, while the program has been used to simulate a wide variety of array designs, this report presents only some sample outputs to illustrate the use of the program. Some results for random arrays, obtained with this program, have already been reported [Blair *et al.* 1994, Blair *et al.* 1997].

The third aim is to give the theory of image-forming, dechirping and the quadrature part (demodulation). While not new, these items of theory, presented here in the context of sonar, are believed to be of interest to readers. As discussed below, these items pertain to real, not just simulated, systems.

The components of the simulated system are discussed in Sections 2 and 3. Section 2 discusses the projector, medium and targets, as well as the transmitted signal. The latter may be either a linear chirp — later to be 'dechirped' — or a short toneburst. Section 3 discusses the simulated receiving array, including the types of receiving element permitted, their placement and the many types of array that can be modelled, including random arrays. Note that the model requires that the total target consist of a small number of point targets. This means that the program is not suited to imaging extended bodies. Section 4 describes the graphical output, which may consist of either three orthogonal 2-D slices through the image or three '1-D slices,' i.e. lines or curves.

Section 5 describes the 'delay-and-add' method of image-forming and methods of demodulation, especially via the quadrature part. The section explains — subject to the model's assumption that the target is a collection of omnidirectional point reflectors — how the method yields an 'image amplitude' that is indeed a good approximation to the total target. This result is obtained even in the near-field, broadband, 3-D-image context. Section 6 discusses the modifications that must be made to the image-forming method when the technique of pulse compression (dechirping) is used.

The case where the received signals are quantised (digitised) is discussed in Section 7. Note that while POINTSPR deals accurately with quantisation in the case of a short toneburst, its treatment of quantisation for the correlated chirp is at best approximate. At this point it is noted that for some situations POINTSPR is not computationally efficient; as a possible means of remedying this an alternative approach, developed elsewhere, is described. Section 8 illustrates the use of the program by giving sample inputs along with the resulting output graphs. Conclusions are given in Section 9.

---

[1] Detailed information on the use of POINTSPR and the related display programs, in their original versions, is contained in the unpublished report: Anstee, S. and Blair, D. [1993], 'Sonar Image Prediction Programs: User's Guide.'

1.2 The Real and the Simulated System

This note is provided to help clarify which aspects of the report apply to the real system and which apply only to the simulated system.

As a preliminary, a quantitative account may be said to describe a *real* (or operational) sonar system if the errors in the description are small—say, a few percent. Given the technical difficulties that prevent a real simulation of the entire system including the target, a simulation with a non-real component can still be very valuable. Two parts of the report can be readily identified as describing aspects of the real underwater imaging system. The theory of these may be of special interest to some readers.

The first part consists of waveforms that are used in practice; these are the subject of Section 2.3.1. Closely connected to this topic are the properties of real transducers, real targets and an attenuating medium. These are briefly discussed in Section 2.1.

The second and more important part consists of three topics that concern the processing of the received signals. They are the image amplitude function (Section 5.1.1, supplemented by 5.2.3.2 and 6.1), demodulation (Sections 5.2.1 to 5.2.3) and dechirping (Sections 6.1 and 6.2.1; the combination of dechirping with demodulation is discussed in Sections 6.2.2 and 6.2.3). In each case an argument is given for the procedure (or alternative procedures) adopted in real processing systems. The procedure is viewed as 'real' because it is actually adopted.[2]

2. Projector, Medium and Targets

2.1 The System Leading up to the Receiving Array

In the simulated system, the transducer system consists of a planar receiving array and an isotropic point *projector* located at the centre of that array (Figure 1).[3] The receiving array is discussed further in Section 3. We take the projector (centre of array) as the origin of coordinates. The displacement of a general point is $\mathbf{r} = (x, y, z)$, with the $+z$ axis ('broadside') directed perpendicular to the array and forwards (Figure 2). Note

---

[2] But interestingly, the argument in the case of the first and third of these topics is based only on a model or simulation of the real system. Thus, the argument for the image amplitude function is based on the target's being a collection of omnidirectional point scatterers. Likewise, the argument for the methods used for dechirping is based on there being no frequency dependence in the attenuation coefficient of the water or in the scattering response of the target. Such are the difficulties of giving a treatment that is consistently real.

[3] Neither the 'planar' nor the 'central' condition need be satisfied in a real imaging system. However in the AMI program, in the systems either built or planned, the array is planar; while the spherical centre of the projector lies somewhat behind the plane of the array.

that an isotropic point projector also satisfactorily models a spherical projector; in that case the equivalent point projector is located at the centre of the sphere.[4]

It is assumed that the projecting transducer is a linear device, that is, the output acoustic amplitude (fluid velocity or pressure) is proportional to the input electrical amplitude (current). It is also assumed that the transducer's sensitivity (constant of proportionality in the above) is independent of frequency. The *input signal*, or electrical current input into the projector, is assumed to be a *chirp* pulse of the form

$$\xi(t) = K_1 \cos\left(2\pi f_c t + bt^2\right) \qquad -\frac{T}{2} < t < \frac{T}{2}$$

$$= 0 \qquad\qquad\qquad \text{otherwise} .$$

(2.1)

Hence, by the two preceding assumptions, the output pressure is also of this form. In Equation (2.1), t is time, T the signal duration and f_c the central frequency, while b/π is the rate of change of frequency, and K_1 is a constant which may be put equal to unity for the purposes of the programs. Thus the signal is a rectangular-envelope, linear-FM pulse [Ziomek 1985]. By inputting the particular case $b = 0$, we obtain a rectangular-envelope, monofrequency pulse, or *toneburst*. Note that to simulate a real system with good range resolution, one needs to use *either* a correlated chirp (i.e. a chirp followed by the dechirping process) *or* an uncorrelated short toneburst.[5] Note also that in a real system, while the input signal may well be a chirp or toneburst, the signal is altered during its passage. Such alteration occurs due to the nonuniform frequency response of the projector, the hydrophone, and any amplifiers and filters, as well as the frequency-dependent absorption by the medium. The program ignores these effects.

The *medium* is assumed to be ideal: it is linear, the speed of acoustic waves c is independent of frequency (equal to 1500 ms^{-1}), there is no attenuation in the medium and there is no scattering (except by the targets). Thus, with the invocation of assumptions from the previous paragraph, the pressure at the point $\mathbf{r}_j$ due to the projector is [Clay and Medwin 1977]

$$p_{\text{pro}}(\mathbf{r}_j, t) = \frac{K_2}{r_j} \xi\left(t - r_j/c\right) ,$$

(2.2)

where K_2 is a constant and $\mathbf{r}_j$ is the location of the j th scatterer.

The total *target* consists of a number of isotropic pointlike scatterers (reflectors) each with an associated 'target strength'. Each reradiates a spherical wave; the amplitude of the wave being radiated at time t is proportional to the target strength and to the pressure of the wave incident from the projector. Thus the pressure of the total scattered wave at a point $\mathbf{R}_n$ (see Fig. 1) is [Clay and Medwin 1977, Urick 1983]

---

[4] Systems built or planned in the AMI program use a projector consisting of portion of a spherical surface.

[5] The program however puts no restriction on the combinations of this kind that are allowed: any length of pulse may be combined with a zero or nonzero value of b and with crosscorrelation or its absence. The choice of appropriate combinations is left to the user.

$$p_{\text{scat}}\left(\mathbf{R}_n, t\right) = \sum_j \frac{a_j}{\left|\mathbf{r}_j - \mathbf{R}_n\right|}\, p_{\text{pro}}\left(\mathbf{r}_j, t - \left|\mathbf{r}_j - \mathbf{R}_n\right|/c\right)$$

$$= K_2 \sum_j \frac{a_j}{r_j \left|\mathbf{r}_j - \mathbf{R}_n\right|}\, \xi\!\left[t - \left(r_j + \left|\mathbf{r}_j - \mathbf{R}_n\right|\right)/c\right],$$

(2.3)

where a_j (a real number) is the *target strength* of the j th scatterer, $\mathbf{R}_n$ is the location of the n th element (see Section 2.2), and Equation (2.2) has been used to obtain the last line. Thus it is assumed that there is no multiple scattering and no 'shadowing' of one target point by another. Clearly it is also assumed that each target strength is independent of frequency and that there is either no phase change upon scattering or a constant phase change of 180° (case of a_j negative). Note that the 'target strength' defined here is related to the more usual 'target strength,' TS, by

$$TS_j = 20\log_{10}\left|a_j\right|,$$

where TS is based on a reference distance of 1 metre (not 1 yard).
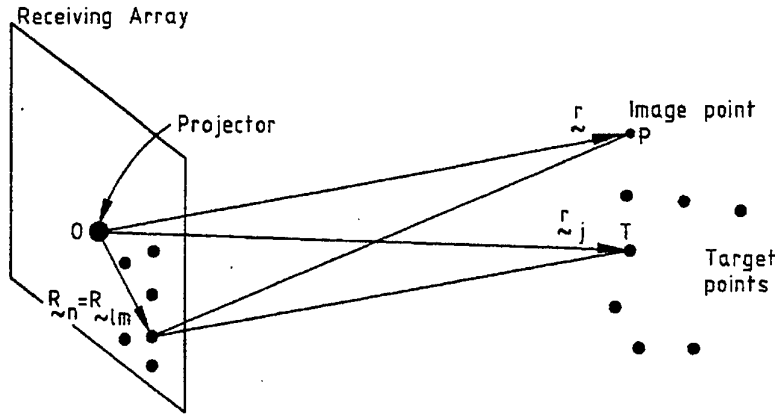
Figure 1: *The overall geometry of the imaging system. The acoustic wave emanating from the projector at O is scattered from a typical target point T and received at the n th array element located at $\mathbf{R}_n$. The points $\mathbf{R}_n$ form a subset of the original grid points $\mathbf{R}_{lm}$ (Section 3.1) in the case of pointlike elements — but a subset of the adjusted grid points in the case of extended elements. The image point P is a typical point at which the image intensity is to be calculated.*

The user may specify any number of point targets between one and 100, thus allowing some degree of approximation to an extended reflector or scatterer.

However, limitations of computer memory and speed have so far made the latter difficult.

2.2 Coordinate Systems

In the right-handed, rectangular system (x, y, z), the $+y$ axis is thought of as upwards. A spherical polar system (r, θ, ϕ) is also defined, in which the elevation θ is the angle above the xz plane and the azimuth ϕ is measured from the $+z$ axis towards the $+x$ axis (Fig. 2). The two systems are related by the equations

$$x = r \cos \theta \sin \phi$$
$$y = r \sin \theta \tag{2.4}$$
$$z = r \cos \theta \cos \phi \ .$$

Thus y (not the usual z) is the polar axis, and latitude θ rather than colatitude is used.
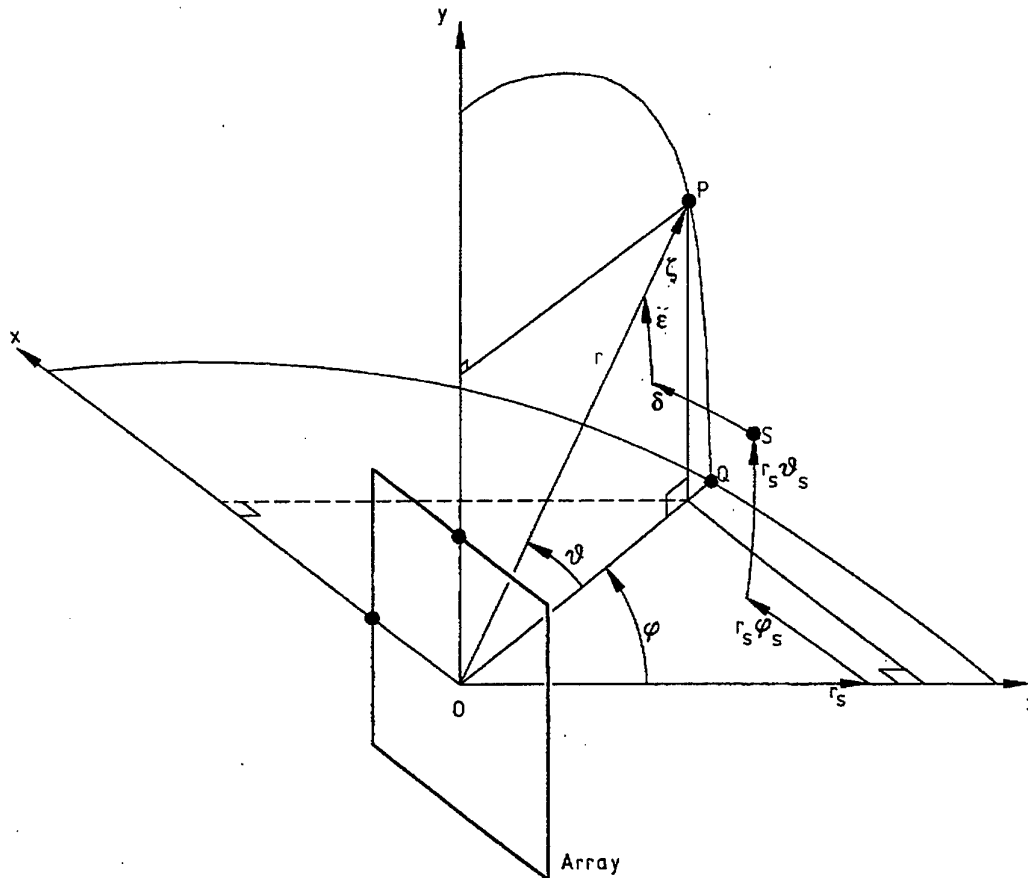
Figure 2: *The rectangular coordinate system (x, y, z) and the spherical polar system (r, θ, ϕ) used to locate a point P. The projector is at O. The two long curved lines are great circles on the sphere $r = constant$ through P. The figure also illustrates the unrotated radially-curved viewgrid. In that context, S is the secondary origin. From S, the curvilinear displacements δ, ε and ζ, in that order, lead finally to position P. (During the δ displacement, r and θ are constant and ϕ changes; and similarly for ε and ζ.)*

When the operational system is image-forming to obtain the image intensity at a given point, that point is referred to as the *image point* (Section 5.1). The coordinates

$\mathbf{r} = (x, y, z) = (r, \theta, \phi)$ *without a subscript* will normally be reserved for the location of the image point. For the location of a target, the subscript j (for the j th target point) is added throughout; thus we write, for example, (r_j, θ_j, ϕ_j). For an element location, capitals are used and the subscript n (for the n th element) is added; thus we write

$$\mathbf{R}_n = (X_n, Y_n, Z_n) = (X_n, Y_n, 0).$$

2.3 Waveforms

We now discuss waveforms that are used in practice, and the extent to which the program's waveform (2.1) successfully models these.

2.3.1 Waveforms Used in Practice

To achieve good range resolution, one strategy is to use a sufficiently short pulse. Because of the two-way path, the relation between range r to the target and round-trip time t is[6] $2r = ct$. A pulse is 'smeared' in time and this leads, in the image of a point target, to a smearing in r. Let T be the duration of the rectangular-amplitude pulse. Let us write the range resolution as $\Delta_0 r$, where $\Delta_0 r / 2$ is the maximum deviation from the centre of the image. Then the range resolution for the two-way system is

$$\Delta_0 r = cT/2. \tag{2.5}$$

In particular, to achieve a 1 mm range resolution, a pulse duration of $1.33\,\mu s$ is required.

The alternative strategy is to use a long chirp together with crosscorrelation (see Section 6). In this report the term 'crosscorrelation' simply implies that the appropriate mathematical relationship exists between three functions: a raw received signal, a transmitted signal and a processed signal. The term does not imply that the processing system concerned (i.e. the program or the operational system) carries out a numerical crosscorrelation, involving a shift operation and a summation of products, to obtain the third of these functions. This apparent discrepancy arises for two reasons. First, in any processing system, the dechirping or 'crosscorrelation' can be done by other processes, in particular by transforming into the frequency domain. Second, in the simulation program, the correlation was done *analytically* before the program was written, so the program contains no 'crosscorrelation routine.'[7]

To obtain the formula for the range resolution in the case of a correlated chirp, we anticipate results from Section 6. From Equations (6.23) and (5.7), for a point target, the image amplitude as a function of range near the peak of the image (at $r = r_1$) is

---

[6] The relationship is approximate, since receiving element and projector are not co-located.

[7] In the simulation program, the 'raw signal' never actually exists as a sequence of numbers; the processed signal is calculated directly.

$$\Pi = \left| \mathrm{sinc}\left[2B\left(r - r_1\right)/c \right] \right|, \qquad (2.6)$$

where B is the bandwidth of the chirped pulse. By considering the first zero of the sinc function, it follows immediately that the distance from the centre to the first zero of the image amplitude is

$$\Delta_1 r = c/2B.$$

Also, the 3 dB width (two-sided) of the image amplitude function is [Steinberg 1976]

$$\Delta_2 r = 0.886 c/2B; \qquad (2.7)$$

this is also taken to be the resolution.

Therefore, with a correlated long chirp, a given resolution is achieved provided that the chirp has *sufficient bandwidth*. From (2.7), for 1 mm range resolution, a bandwidth of at least 0.664 MHz is required.

For the short pulse as well as for the correlated chirp, the requirement needed to achieve a given resolution can be expressed in terms of the bandwidth B (rather than T). Thus the spectrum of the toneburst has a bandwidth, as measured by its 3 dB width, of

$$B' = 0.886/T;$$

hence the resolution (2.5) may be expressed as

$$\Delta_0 r = 0.886 c/2B'. \qquad (2.8)$$

Comparing (2.7), we see that the resolution is related to bandwidth by the identical formula for the two types of pulse, but with B interchanged[8] with B'.

Medical ultrasound [Hill 1986, Webb 1988, Fry 1978] always uses the technique of a short pulse with no crosscorrelation. A typical pulse [Robinson and Wing 1984] is shown in Figure 3(a); this pulse is approximately a sinusoid (carrier wave) multiplied by a smooth envelope that extends for about 12 half-cycles, but for which the amplitude trails off as one moves away from the centre in either direction. Note that this modulated pulse is not the shape of the input electrical signal, which typically is simply a half cycle of a sine curve (inverted 'U' shape). Rather, the modulated pulse is the shape of the electrical signal output by the receiving element after the signal has been distorted by the respective transfer functions of the transmitting transducer, the target, and the receiving transducer. *The system is equivalent, however, to a system in which:* (i) the pulse that is finally output (e.g. the pulse in the figure) is also the shape of the input, and (ii) the transducers act without distortion [D.E. Robinson, private communication]. Thus the pulse that Figure 3(a) typifies is the *'effective'* input signal $\xi(t)$.

---

[8]Note that this identity requires that B', the 3 dB width, be used for the toneburst. A little thought shows that B for the chirp is also formally the 3 dB bandwidth, due to the nature of the step function.
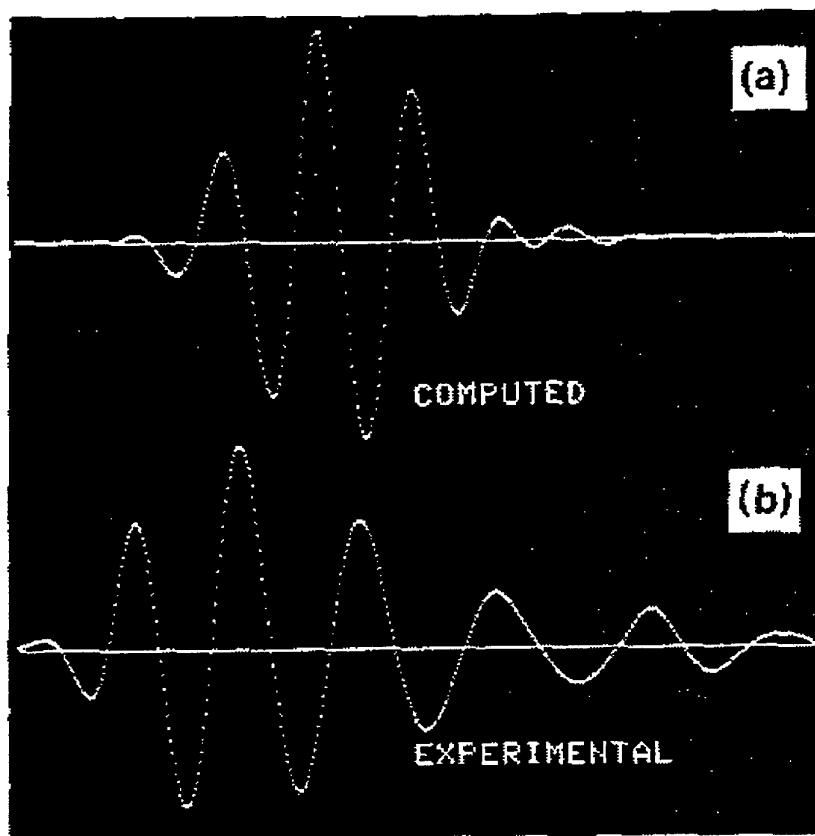
Figure 3: *Pulses for medical ultrasound. (a) Model pulse used in computer generation of beam response. The pulse represents the voltage echo response from a very short pulse electrical input. As far as can be ascertained, the pulse represents the true point echo response. (b) Actual experimental voltage echo off a nylone line in water. The ringing is due to the line's not being a point. Figure reproduced from Robinson and Wing [1984].*

It has been found [D.E. Robinson, private communication] that a good signal for modelling such real signals in medical ultrasound is

$$\xi(t) = K_1 \cos\left[(1/5)2\pi f_c t\right] \cos\left(2\pi f_c t + \varepsilon\right) \qquad |t| \le T/2$$
$$= 0 \qquad\qquad\qquad\qquad\qquad\text{otherwise}$$

(2.9)

with $T = 5/(2f_c)$, and ε a phase constant. This signal is illustrated in Figure 4 for the cases $\varepsilon = 0, -\pi/2$. For the case $\varepsilon = 0$, the signal contains five half-cycles. Note that the envelope (the first of the two cosine factors) falls linearly to zero at the two ends, but that the signal $\xi(t)$ itself falls to zero as $|t - t_{end}|^2$. Robinson also notes that in medical ultrasound practice, on occasions the effective signal $\xi(t)$ falls to zero linearly, i.e. as

$\left| t - t_{end} \right|$, as in the case $\varepsilon = -\pi/2$; sometimes (when $\varepsilon = 0$) $\xi(t)$ drops to zero more smoothly than linearly; but it never drops to zero as a step function.

Waveforms $\xi(t)$ that fall to zero linearly at the ends, i.e. that end with a discontinuity in slope, will be called *ramp-end* waveforms. Waveforms that (contrary to medical practice) have a step function at each end will be called *step-function-end* waveforms. In practice neither of these waveforms is realisable due to bandwidth limitations on the transducer system. For example, in the ramp-end waveform the discontinuity in slope is smoothed out to some degree. These concepts are applied to the program's waveform (2.1) in Section 2.3.2.
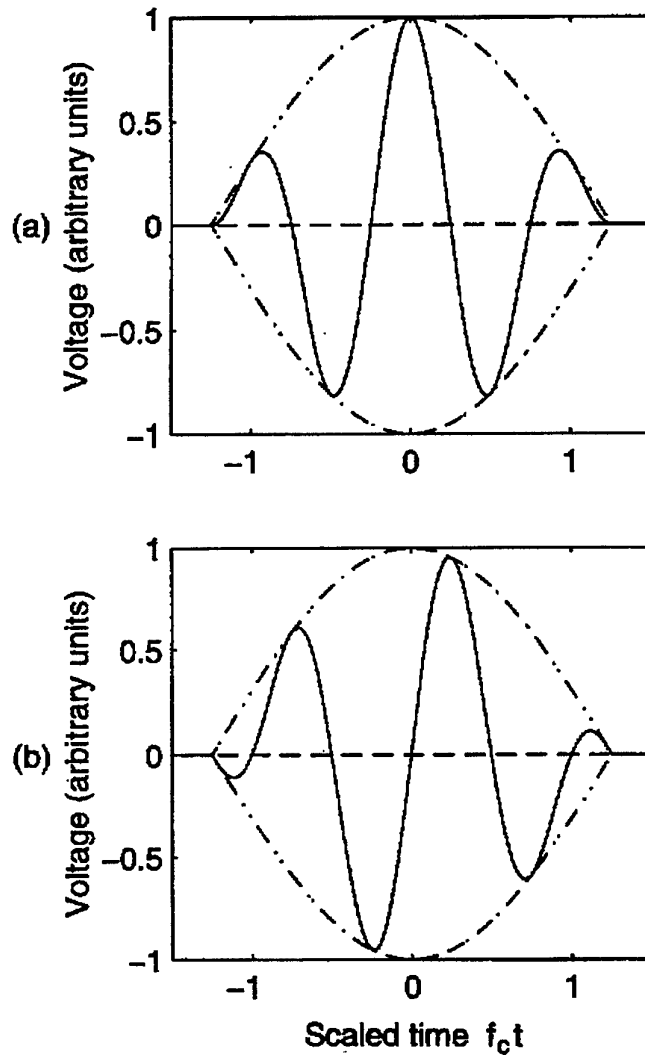
Figure 4: Simple model signals for medical ultrasound, approximating Figure 3(a) and given by Equation (2.9). (a) Case of phase constant $\varepsilon = 0$. (b) Case $\varepsilon = -\pi/2$.

In *sonar*, a correlated signal—either a chirp or a coded signal—is most often used. A system suitable for the proposed operational sonar is as follows [D.E. Robinson, private communication]. The electrical input signal is a long chirp (many cycles) with a rectangular envelope. The transducers, however, have a frequency response that falls off, in a way that is gradual but with some irregularities, as the frequency moves towards the extreme frequency values of the input chirp. The frequency response can be extremely small at those extreme frequencies; in fact this is what normally happens without any intentional design. The resulting system is *equivalent* to distortionless transducers (and targets) combined with an input signal of the form

$$\xi(t) = K_1 \, a(t) \cos\!\left(2\pi f_c t + bt^2 + \varepsilon\right) \qquad |t| \le T/2$$
$$= 0 \qquad\qquad\qquad\qquad \text{otherwise,}$$

(2.10)

where ε is a phase constant and, in a model that ignores the irregularities, the envelope $a(t)$ is a smooth function, dropping to practically zero at the extreme times $t = \pm T/2$. Such a waveform, having a gaussian envelope

$$a(t) = \exp\!\left[-t^2/\left(2\sigma^2\right)\right],$$

(2.11)

is shown in Figure 5.
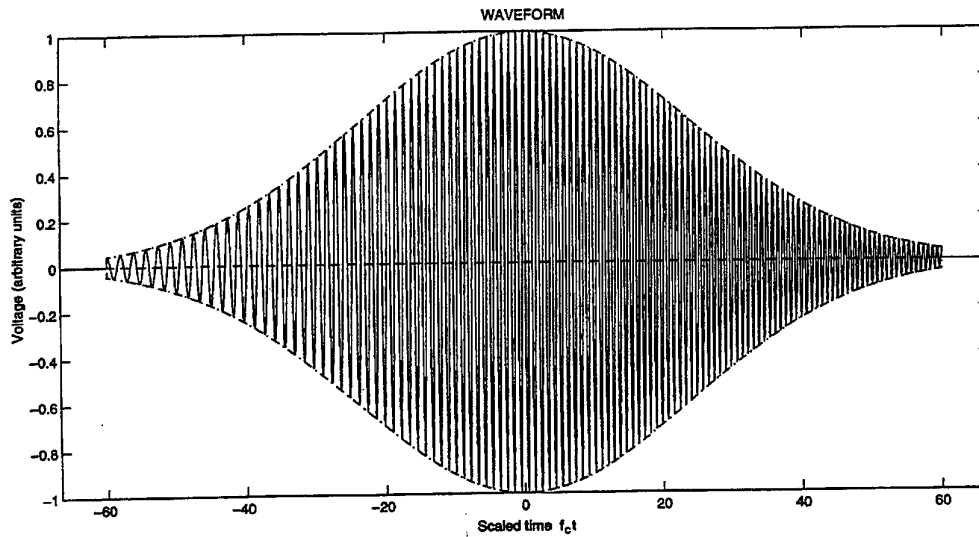
Figure 5: Chirp waveform with a cutoff gaussian envelope, given by Equations (2.10) and (2.11). The parameters are as follows: number of cycles $= f_c T = 120$; standard deviation $\sigma = 0.4(T/2)$; $\varepsilon = 0$; bandwidth $= 1.0 f_c$ or, from Equation (6.22), $b = 1.0 \pi f_c / T$.

2.3.2 Realistic Modelling with the Program

In both the uncorrelated and correlated cases, the program's waveform differs from the waveforms normally used in practice (Section 2.3.1). The program is still useful, however, in modelling many of the effects, for example, the dependence of range resolution on bandwidth—though not in modelling the detailed shape of the range sidelobes.[9]

However, it is desirable to avoid waveforms that lead to *artificial features* in the image—as will now be discussed.

A key difference between the program's waveform (2.1) and the waveforms used in practice, is the step-function ending of the *envelope* in (2.1). Likewise the waveform $\xi(t)$ *itself* has a step-function ending unless the cosine has a zero at each end. Figure 6 (a and b) shows two such ξs, with and without step-function endings.

Consider first the case of an *uncorrelated* waveform (short toneburst). Then it turns out that step-function endings of $\xi(t)$, besides being at variance with practice, lead to an effect on the image intensity function to do with the quadrature part of the image (Section G.4 in Appendix G)—an effect that is unfortunate, though it can be lived with. For this reason, preferably *the parameters in Equation (2.1) (b and T, for a given f_c)* *should be chosen so as to produce a node at each end*. If this recommendation is followed, in the case of a toneburst it is required that

$$2\pi f_c(T/2) = m\pi + \pi/2 \qquad (2.12)$$

where m is an integer. Actually the further case

$$2\pi f_c(T/2) = m\pi \qquad (2.13)$$

is permitted; in this case the program models the waveform given by (2.1) *but with sine replacing cosine*. (This result follows from Sections 5 and 6, particularly Section 5.2.) Such a waveform is shown in Figure 6(c).

---

[9] Such results, obtained for the program's waveform, are still of theoretical interest, however.
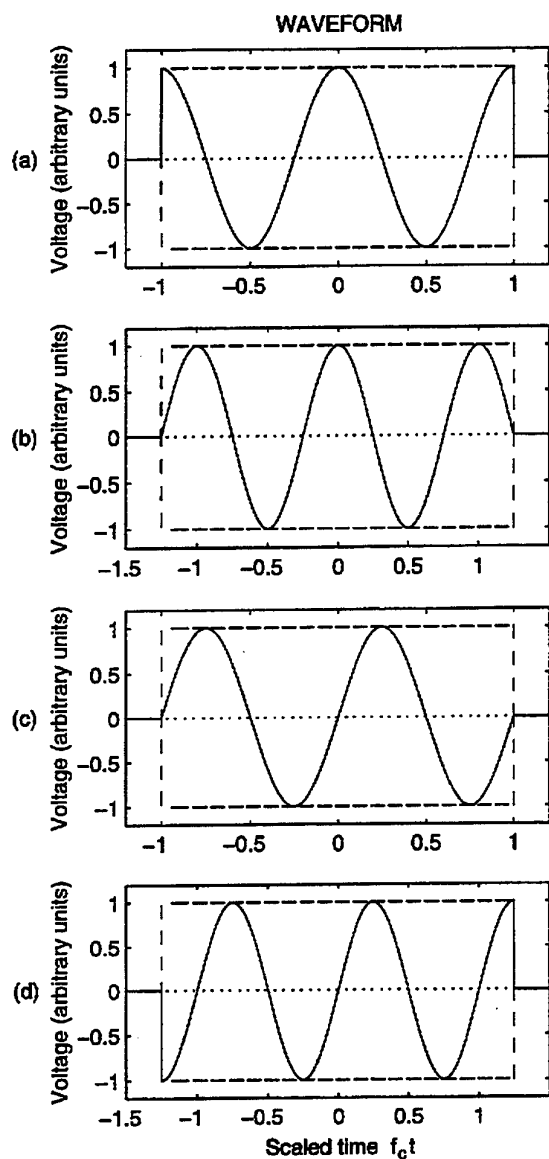
Figure 6: *Short tonebursts illustrating realistic modelling with the program. All are given by Equation (2.1) with b = 0, but with sine possibly replacing cosine. (a) Cosine, $f_c T = 2$. Step-function ending. (b) Cosine, $f_c T = 2.5$. Ramp ending. (c) Sine, $f_c T = 2$. Ramp ending. (d) Sine, $f_c T = 2.5$. Step-function ending. The pulse shapes (b) and (c) are typical of what might be used in the operational system if an uncorrelated toneburst (rectangular envelope) were used (see Section G.3).*

In the case of a *correlated* waveform (chirp), in general it is not necessary to impose nodes at the ends of the waveform, as Appendix A shows.

3. Receiving Array; Elements

Transducers have been discussed by Burdic [1991] and Kino [1987]. Arrays of transducers have been discussed by Steinberg [1976], Burdic [1991], Ziomek [1985], Kino [1987], and more briefly by Urick [1983].

3.1 Array Grid

In the simulation by POINTSPR, the construction of the receiving array proceeds in steps. First, a square grid of $M \times M$ *sites* is set up, with spacing d and centre at the origin, which coincides with the projector. The displacement of the (l, m) th grid point is

$$\mathbf{R}_{lm} = \left[l - (M+1)/2\right]d\hat{x} + \left[m - (M+1)/2\right]d\hat{y} \qquad \begin{array}{l} l = 1,\dots, M \\ m = 1,\dots, M, \end{array} \tag{3.1}$$

where M is the number of grid points along a side, and $\hat{x}$ and $\hat{y}$ are unit vectors in the x and y directions respectively. The grid is considered to occupy a region of dimensions $Md \times Md$, called the grid region, which therefore extends $d/2$ beyond each of the four end lines of grid points. This grid region may be considered as made up of cells, each of dimensions $d \times d$ and having a grid point at its centre (Fig. 7). The physical significance of the grid points (3.1), in the case of pointlike elements, is that the sensor elements are later to be placed at some, but in general not all, of these points.
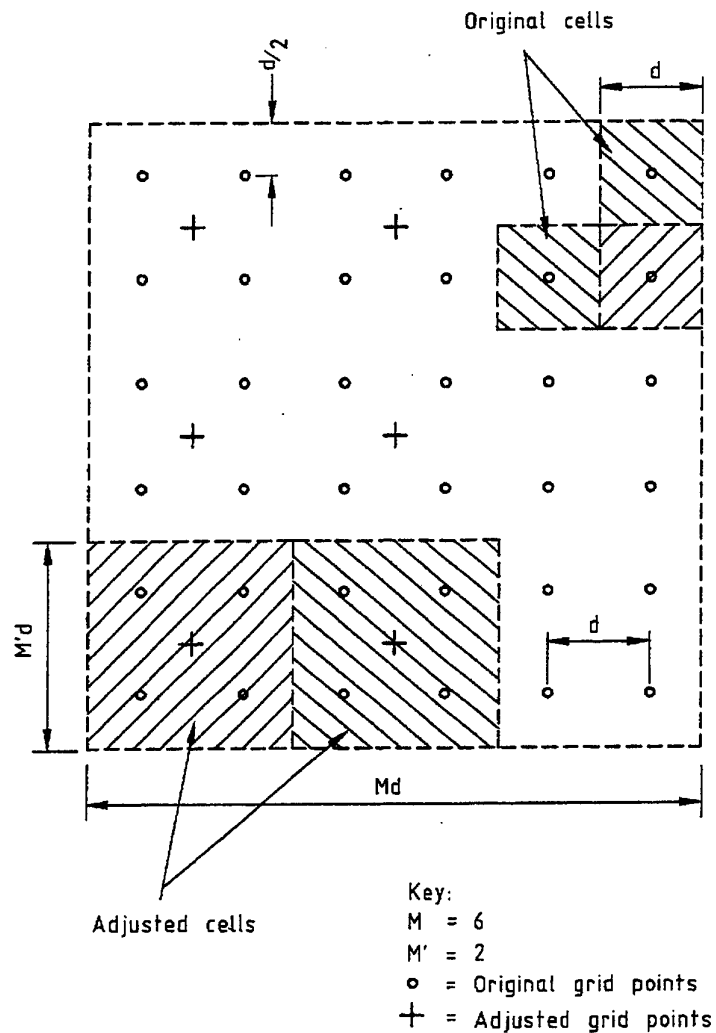
Figure 7: *Showing the construction of the adjusted grid from the original grid. Construction proceeds via the formation of the original cells, which exactly cover the grid region and which have an original grid point at their centre. These cells are then combined to form adjusted cells; the centres of these give the adjusted grid points. In the figure, adjusted grid points towards the right-hand edge of the grid are omitted.*

Two main types of element are allowed in the simulation: *pointlike* and *solid*. There is also a third type, *tilelike*, for which little use has been found. In the case of the solid and tilelike elements (jointly referred to as *extended*), the element must be square and oriented with its principal axes parallel to the x and y axes. The length of its side, chosen by the user, must be of the form $M'd$, where both M' and M/M' are integers.

Circular as well as square elements are of interest for a real system. The square elements in the simulation give at least a semiquantitative estimate of the effects, say on the point spread function, of using circular elements. (The *point spread function* is the image of a single point target.)

In the simulation, extended elements are accommodated by introducing an *adjusted grid*. The latter is defined by re-subdividing the grid region (of dimensions $Md \times Md$) into cells that are in general larger than before. The adjusted cells have dimensions $M'd \times M'd$ and there are M/M' adjusted cells along a side (Fig. 7). The *adjusted grid points* are at the centres of the adjusted cells. In the case of pointlike elements, $M' = 1$ and the adjusted grid and the original grid coincide.[10]

The physical significance of the adjusted cells, for extended elements, is that elements are later to be placed to neatly fill some, but in general not all, of the adjusted cells. The program requires that in any one simulation all the elements placed *must be identical*, both in type (point, solid or tilelike) and in size. Furthermore, *an element may only be placed with its centre at an adjusted grid point*. Then clearly the extended elements never overlap, though they may have a side or a corner in common. For solid elements, once the adjusted grid is set up, the original grid is 'thrown away,' i.e. not used.

3.2 Types of Element

As stated above, in the model there are three types of elements. Each element is assumed to respond instantaneously to the pressure at the points on its front face, and to respond to those points with equal weight. Furthermore this pressure is assumed to equal the value that the scattered pressure p_{scat} would have in the absence of the array. Moreover the response is taken to be linear, i.e. the voltage response is proportional to the pressure.

3.2.1 Pointlike Elements

If the elements are pointlike, the front face is modelled by a single point. Then the output voltage of the nth element, located at $\mathbf{R}_n$, is given simply by

$$E_n(t) = K_3\, p_{scat}(\mathbf{R}_n, t), \tag{3.2}$$

where K_3 is constant. Thus the sensitivity is independent of both frequency and direction of arrival.[11] Substituting from Equation (2.3), we find that

---

[10] Note that the adjusted grid points do not necessarily lie at positions occupied by original grid points. A consideration of Figure 7 shows that, for M' odd, the points do so coincide, but for M' even, each adjusted grid point lies at the centre of the square formed by four original grid points adjacent to each other.

[11] By contrast, extended elements possess both sensitivities, but only as a result of their structure: for, at the level of their smallest component (an elementary area dS in the case of a

$$E_n(t) = K_2 K_3 \sum_j \frac{a_j}{r_j |\mathbf{r}_j - \mathbf{R}_n|} \xi\left[t - \tau(\mathbf{r}_j, n)\right], \tag{3.3}$$

where

$$\tau(\mathbf{r}_j, n) = \left(r_j + |\mathbf{r}_j - \mathbf{R}_n|\right)/c \tag{3.4}$$

is the round-trip time for a pulse travelling from the projector to the n th element via a scatterer at $\mathbf{r}_j$. The formulae for the voltage under various conditions, beginning with Equation (3.3), are summarised in Section 6.7, along with equations for the image amplitude.

3.2.2 Solid Elements

3.2.2.1 General

If the elements are solid, each element is a square of side $M'd$ and acts as a square piston. The output voltage is

$$E_n(t) = K_5 \int_{S_n} p_{\text{scat}}(\mathbf{R}, t) \, dS, \tag{3.5}$$

where the surface integral over the variable $\mathbf{R}$ is across the square front face S_n of the element and K_5 is a constant. Substituting for the pressure from (2.3), and noting that the pressure is to be evaluated at $\mathbf{R}$, not $\mathbf{R}_n$, we find

$$E_n(t) = K_2 K_5 \sum_j \frac{a_j}{r_j} \int_{S_n} \frac{1}{|\mathbf{r}_j - \mathbf{R}|} \xi\left[t - \left(r_j + |\mathbf{r}_j - \mathbf{R}|\right)/c\right] dS. \tag{3.6}$$

In addition, for solid elements the model assumes that the point targets lie in the far field of the *element*. As shown in Appendix B, the result becomes[12]

$$E_n(t) = K_2 K_5 \sum_j \frac{a_j}{r_j |\mathbf{r}_j - \mathbf{R}_n|} \int_{S_n} \xi\left\{t - \left[r_j + |\mathbf{r}_j - \mathbf{R}_n| - \frac{(\mathbf{r}_j - \mathbf{R}_n) \cdot \mathbf{R}'}{|\mathbf{r}_j - \mathbf{R}_n|}\right]/c\right\} dS' \tag{3.7}$$

where the variable of integration has been changed from $\mathbf{R}$ to $\mathbf{R}' = \mathbf{R} - \mathbf{R}_n$. The condition of validity of the far-field assumption is

$$\min_j \left[\frac{r_j}{\cos^2(\mathbf{r}_j, \hat{z})}\right] \gg \frac{(M'd)^2}{\lambda_c},$$

where $\lambda_c = c/f_c$ is the 'central' wavelength and the minimum is taken over the targets. [This result follows from the fact that, for a square array of dimension L, the

---

solid element, or a subelement in the case of a tilelike element), there is neither directional nor frequency sensitivity.

[12] This result, which is for a general b but with an uncorrelated pulse, might appear to be of no practical use. However it later forms the conceptual basis for deriving the result for the correlated chirp (Y replacing ξ). The latter *is* useful.

near field-far field transition range is L^2/λ at broadside, or $(L\cos\theta)^2/\lambda$ in a direction at angle θ from broadside.]

Note that, because the system does not apply any relative time-delays to the parts within one element, the solid element is *unsteered*. Note also from (3.5), that solid elements are unshaded: in the context of apodisation (Section 3.4) there is no change of weight as one moves across the parts of the element. The weight is simply governed by the position that the centre of the element occupies in the array.

3.2.2.2 *Case of Toneburst*

For a solid element combined with a toneburst, as shown in Section B.2, a closed expression (3.11 below) can be obtained for the voltage. Recall that the (far-field) directivity $D(u, v, f)$ of an aperture is defined in terms of a plane wave of single frequency f arriving at the aperture, from a direction given by (θ, ϕ), or alternatively by the direction cosines (u, v) (see below). The directivity is the voltage response, normalised to the response at broadside (no internal steering being applied to the element). The directivity of a square piston is [Steinberg 1976, Ziomek 1985]

$$D_s(u, v, f) = \text{sinc}(fc^{-1}Lu)\,\text{sinc}(fc^{-1}Lv),\tag{3.8}$$

where the subscript s is for square, L is the side of the piston and

$$\text{sinc}(x) = \sin(\pi x)/\pi x.$$
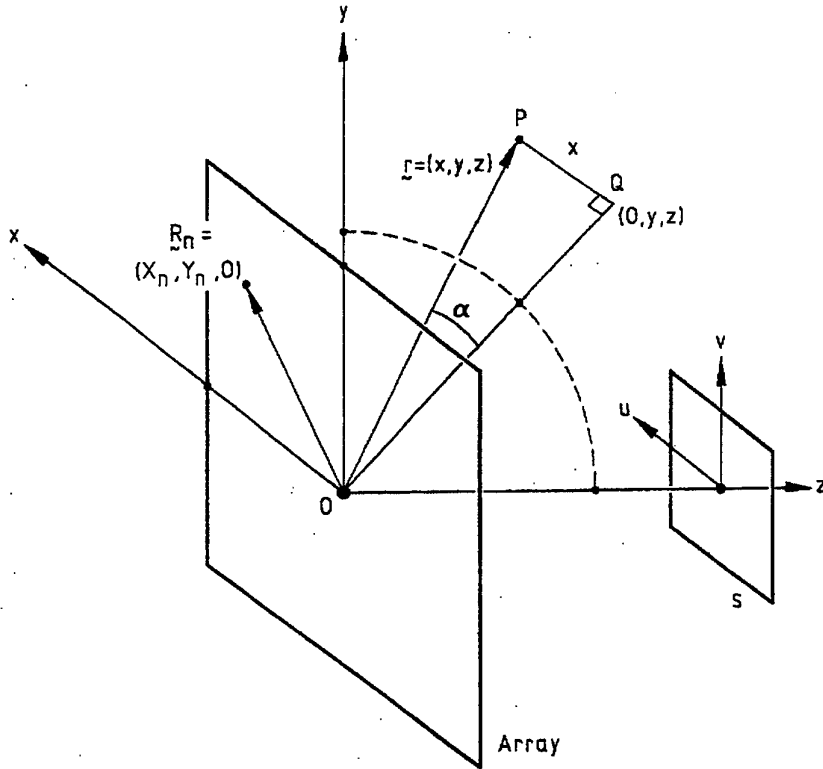
Figure 8: *Illustrating the direction cosines u and v. P, the image point, when projected onto the yz plane, yields Q. The first direction cosine is $u = \sin\alpha$, where α is shown. To first order in angle, u and v are angular coordinates (see text); to the same order, the directions of increasing u and v are as shown.*

In the context of the nth element receiving a signal from the jth target, the direction cosines (each being a measure of displacement from broadside) are

$$u_{jn} = (x_j - X_n)/|\mathbf{r}_j - \mathbf{R}_n|, \qquad v_{jn} = (y_j - Y_n)/|\mathbf{r}_j - \mathbf{R}_n|. \tag{3.9}$$

Of these, for example the first is the sine of the angle that the vector from the centre of the nth element to the jth target makes with the yz plane (along with a sign convention); equivalently, it is the cosine of the angle made with the $+x$ axis (Fig. 8). Much use is made of these direction cosines in radioastronomy [e.g. Perley *et al.* 1989]. The directivity is then

$$D_s(u_{jn}, v_{jn}, f_c) = \text{sinc}(f_c c^{-1} M' d u_{jn}) \text{sinc}(f_c c^{-1} M' d v_{jn}). \tag{3.10}$$

The output voltage (Appendix B) is

$$E_n(t) = K_2 K_5 (M'd)^2 \sum_j \frac{a_j}{r_j \left| \mathbf{r}_j - \mathbf{R}_n \right|} D_s\left(u_{jn}, v_{jn}, f_c\right) \xi\left[t - \tau\left(\mathbf{r}_j, n\right)\right]. \tag{3.11}$$

It must be emphasised that this result holds only for a toneburst ($b = 0$). The result given by (3.11) and (3.10) is used explicitly in the computer program. It is of interest to note that Equation (3.11) is the same as the result (Eqn 3.3) for point-like elements except that K_3 is replaced by $K_5(M'd)^2 D_s\left(u_{jn}, v_{jn}, f_c\right)$. The Equation (3.11) is subject to a further pair of conditions, given as Equation (B.7).

3.2.3 Tilelike Elements

The program allows a third type of element, the 'tilelike element.' Basically it is a square array of point elements (subelements), with the additional feature that internally the element is *far-field steered*. These elements are described in more detail in Appendix C.

The usefulness of tilelike elements is problematical. They were included in the program because of a certain idea that occurred to the working group, namely, that in the operational imaging system, it might be possible to save considerable computing time by doing some of the image-forming on a far-field assumption. This image-forming would occur preferably just behind the transducer elements, and would possibly involve the fast Fourier transform. But the idea was later seen to face a difficulty, because it has the consequence that the voltage $E_n(t)$ (Eqn 3.11) depends on the image point (as discussed in Section 3.2.3). To be more precise, that voltage depends on the *direction* of the image point $\mathbf{r}$ from $\mathbf{R}_n$. Thus, taken literally, the element would have to output very many voltage streams $E_n(t)$, one for each direction to the image point. Overall the idea seems to require more, not less, computing. And immense computing power would have to be placed in the wet end.[13]

Nevertheless, the tilelike elements should be useful for a *different* purpose, namely, for simulating some arrays of very many elements. For example, under conditions appropriate to the mine imaging situation, to calculate a planar image of a $\lambda/2$ array of 2000×2000 point elements leads to prohibitive problems of storage as well as of computing time. It is expected that the simulation becomes manageable while remaining accurate, if the array is replaced by (say) a 40×40 array of tilelike elements, each of 50×50 subelements.

---

[13] The idea of a tilelike element might become viable if reinterpreted so that the element is to produce an output for only a small number of directions. It can be argued that only such a small number is needed, since the element, being small, has a much larger resolution cell than the array. This interpretation uses the concept of a 'low-resolution image' [D.E. Robinson, private communication].

3.3 Placement of Elements

As noted previously, the elements may be placed (centred) on adjusted grid points only. The array configuration is specified by assigning an occupation number of 1 (occupied) or 0 to each adjusted grid point. The program user specifies these occupation numbers by issuing a list of instructions to be applied sequentially to the initially empty array.

A facility is included to view a graphical representation of the element pattern.[14]

3.4 Shading

The shading (weighting, apodisation) applied to each element n is the weight w_n in the image-forming Equation (5.4) below. Commonly used shadings are discussed by Steinberg [1976] and Ziomek [1985]. In the program POINTSPR, each shading is specified by a formula, which gives the weight in terms of the location of the *centre of the element* with respect to the centre of the array. The main shadings available to the user are listed below.[15] In these formulae, $\mathbf{R} = (X, Y)$ is the location of the element and $a = Md/2$ is half the side of the array grid, while α (real) and n (integer) are parameters whose values are freely chosen by the user.

The shadings are:

1. unshaded	$w = 1$		
2. gaussian	$w = \exp(-R^2/2\alpha^2)$		
3. edge power cosine	$w = \cos^n(\pi R/2a)$	$n = 1,2,3,\ldots$	$R < a$
	$= 0$		$R \geq a$
4. xy power cosine	$w = \cos^n(\pi X/2a)\cos^n(\pi Y/2a)$	$n = 1,2,3,\ldots$	
5. pyramid	$w = 1 - \max(|X|,|Y|)/a$		(3.12)
6. edge conical	$w = 1 - R/a$	$R < a$	
	$= 0$	$R \geq a$	
7. xy triangular	$w = (1 - |X|/a)(1 - |Y|/a)$		
8. corner power cosine	$w = \cos^n(\pi R/2\sqrt{2}a)$	$n = 1,2,3,\ldots$	

The weights as expressed in (3.12) are not normalised (see Section 4.3.1.) There are two ways of achieving an array with circular symmetry.[16] One is to remove the elements for which $R \geq a$ by using the instruction 'ring.' The other is to use one of the two

---

[14] All programs in this report are in FORTRAN except where otherwise stated. No allowance is made for the fact that, due to the presence of the spherical projector, there are sites near the centre of the array that cannot be occupied by a receiving element. The user can take account of this by explicitly removing all such elements.

[15] It is a simple matter to add more shadings to the program.

[16] That is, circular symmetry apart from the relatively small effect of the *internal* square structure of the grid.

shadings described as 'edge.' Note that in the case of extended elements the whole element is given a weight based on the position of the *centre* of the element.

3.5 Preconfigured Tile

The purpose of a preconfigured tile is to enable a given spatial pattern of elements to be repeated, with or without rotation, in the course of building up the pattern of the entire array. Such replication in the operational array might lead to considerable cost savings in construction. In a real array, if the aperture is to be *filled* by replication of a single tile, that tile may be triangular, square or hexagonal. In the simulation, only square tiles are treated.

The program requires that the size of the square tile is $M''(M'd)$, where both M'' and $(M/M')/M''$ are integers and $M'd$ is the adjusted cell size (see Section 3.1). The tile is considered to be a *grid of adjusted grid points*, $M'' \times M''$ of them; at each such point an element may be placed. The word 'adjusted' will henceforth be dropped.
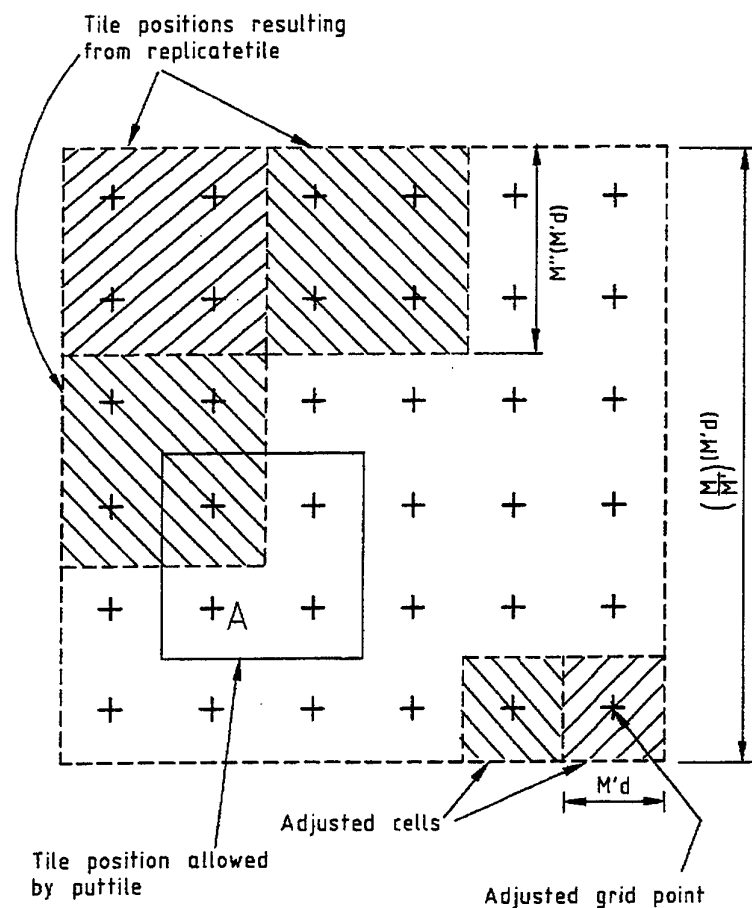
Figure 9: *Illustrating the allowed size of a preconfigured tile, and the placement of such tiles on the adjusted grid. (Point A is explained in the text.)*

The *internal configuration* of each preconfigured tile is input into the program in a similar way to that of the array (Section 3.3), except that the instructions are limited to a subset of the array instructions.

The subsequent *placement of copies* of the tile on the *array grid* is carried out by two further instructions as follows. The instruction 'puttile' essentially causes one copy of the tile to be placed, without rotation, at a location on the grid (Fig. 9). The instruction 'replicatetile' causes the array grid region to be subdivided into squares of size equal to that of the tile, and a copy of the tile to be placed in each square. Options exist to cause

the tile to be rotated, either randomly, or in uniform steps of 90°, as one moves through the array.

Note that the array, once formed, is simply an array of *elements*: thus the time-delays and shadings that are applied are determined by the location of the centre of the element, not the centre of the tile.[17]

3.6 Random Arrays

Random arrays have been discussed by Steinberg [1976], while Steinberg and Subbaram [1991] have discussed ways of processing the output from random arrays to produce enhanced images. The feature that limits the performance of random arrays is the clutter, or distant sidelobes, in the point spread function; these are discussed within Appendix D.

In the context of the program POINTSPR, the instructions mentioned in Section 3.3 enable the construction of simple and more complex random arrays.

To implement randomness, the program uses the subroutine RAN3, proposed by Knuth [1981, Sections 3.2, 3.3] and described by Press *et al.* [1986, p. 198]. RAN3 generates a sequence of floating-point numbers, which (hopefully) are distributed uniformly over the interval $(0,1)$ and are statistically independent of each other. Following the generation of each random number, the routine has in store a vector of 57 seed numbers from which to generate the succeeding random number and seed vector. An original seed vector was calculated, by a special procedure of RAN3, at the first running of the random facility of POINTSPR.

The random number generator used (RAN3) is a novel one. Initially we used a different subroutine RAN1, described also by Press *et al.*, but abandoned it when, in the course of Monte Carlo simulations, quite strong correlations in the supposedly random sequence turned up. Actually Press *et al.* discuss at length the dangers of using 'any old' random number generator, and give various steps that can be taken to avoid such pitfalls. It was therefore most surprising that one of their recommended routines failed in the present context. This may serve as a salutary lesson.

Further facilities in POINTSPR will now be discussed. Consider the situation where the user makes repeated runs with the same *generic array*, that is, the same specification of an array in probabilistic terms. The user has the choice between generating the same *specific array* (definite arrangement of elements) on each occasion or generating a different specific array on each occasion. This is done, in the first case, by using always the original seed vector, and in the second case, by using the most recently generated seed vector.

The user may wish to generate a specific array with a given number of elements, but not an array that by chance has some worse-than-usual property, such as a particular distant sidelobe that is very high. Therefore he or she wishes to generate one or more specific arrays, test each by means of one or more runs, and save one specific

---

[17] This contrasts with the situation that obtains with extended elements in relation to their component parts, at least in respect of shading (Section 3.4).

array that is satisfactory. To cater for this need, on each run, the user may save in a file the specific array configuration. Many such files may be saved. Later, the array to be run may be taken from such a *specific array file*.

For many purposes it is useful to calculate averages of beam patterns over a number of different specific arrays generated by the same generic array, in other words to do *Monte Carlo* simulations. The simulation program has this facility; it is discussed in Appendix D.

A proposal that has been made to save costs in the operational system is to use a single preconfigured tile (Section 3.5) with four orientations randomly chosen, to approximate a truly random array having the same total number of elements. POINSPR has been used to study the performance of this approximate array [Blair, Thompson and Anstee 1997]. In a typical case it was found that the use of a tile raised the clutter level by 5 dB. While this loss of performance is not too bad, the result suggests that it is worthwhile instead to build the array from *two* distinct preconfigured tiles.

4. Image; Viewgrid

The image-forming process, by which an image is obtained from the voltage streams in many real systems as well as in POINTSPR, is described in Section 5.1 and 5.2. In the case of the simulation, its significance is easy to see: the total target is a distribution of target strengths through 3-D space, and the image is the program's estimate of that distribution obtained from the voltages.

The image-forming process may be thought of as the focussing of the hydrophone array, in software, on a set of points r in space, called 'image points', at which the total image amplitude $\Pi_t(r)$ is calculated.

The grid of image points $r = (x, y, z)$ on which the image amplitude function is actually calculated and subsequently displayed is called the *viewgrid*. The viewgrid consists either of points along a line or curve ('linear graph') or points over some surface ('planar graph'). Actually, the viewgrid consists of three such sets of points at right angles to each other. The program POINTSPR outputs to a file an array, consisting of the amplitudes at the viewgrid points; later this array can be plotted by using the appropriate plotting routine: PLINE, PLPLANE or PLMESH.

Usually the actual quantity plotted is the decibel measure of the intensity Π_t^2; optionally the amplitude Π_t (nonlogarithmic) may be plotted. In the case of mesh plots it is most common to plot the amplitude.

4.1 Linear and Planar Graphs

Time and storage constraints preclude the calculation of the amplitude over a three-dimensional grid. Instead, the graphs may be linear (1-D grid, plotting routine PLINE) or planar. In *linear graphs*, a plot of image amplitude versus position along some line or curve is drawn. Also — in this linear case only — the option is available to plot several amplitude profiles on the one graph. In *planar graphs*, over some surface S_0 the amplitude is represented in either of two ways, at the user's choice. The first representation (PLPLANE) is via *false colour*, yielding essentially a contour plot. In the second alternative (mesh plot, PLMESH), a surface S_1 is presented (essentially in perspective), whose height above a base plane S_0 represents the amplitude at the point in S_0 directly below. 'Hidden' lines have been removed.[18]

4.2 Viewgrids

4.2.1 Introduction

A number of viewgrid types are available. In every case these points form a regular grid, usually curved. In place of the old coordinates (x, y, z), the grid is described by some new set of coordinates, which we shall write symbolically as $(\delta, \varepsilon, \zeta)$. The graphs are plotted with the latter coordinates as axes. The new origin, or *'secondary origin'* $(\delta, \varepsilon, \zeta) = (0, 0, 0)$, is usually chosen either at the first target point or at the centre of the viewgrid; very often these coincide.

Note that $(\delta, \varepsilon, \zeta)$ is the *generic* name for the new set of coordinates, so that various options for the choice of $(\delta, \varepsilon, \zeta)$ are available. Very occasionally $(\delta, \varepsilon, \zeta)$ are cartesian coordinates ('flat' viewgrid type, Section 4.2.3); but most commonly the set $(\delta, \varepsilon, \zeta)$ is given by Equation (4.1) below, and is a modification of spherical polar coordinates.

Note also that a planar graph involves the mapping or 'projection' of a surface that is in general curved, onto the plane surface of the paper; hence distortion occurs. For example, the first of the three plots is a plot of the intensity at points (δ, ε) with ζ constant. The plot involves setting up two *cartesian* axes on the paper, labelled δ and ε, and plotting each point (δ, ε) in cartesian fashion. Section 8 gives examples of linear and planar graphs.

Further properties of general viewgrids are given in Section 4.2.2.2.

---

[18] PLMESH is written in MATLAB as a script file (M-file).

4.2.2 Unrotated Case of Radially-Curved Viewgrid

Of the viewgrid types, the *'radially-curved'* type is used almost exclusively. This type is subdivided into the *'unrotated'* and *'rotated'* options; the present Section 4.2.2 concentrates on the unrotated option.

4.2.2.1 Radial Curving

For the unrotated, radially-curved viewgrid, the new coordinates $(\delta, \varepsilon, \zeta)$ are *essentially the spherical polar coordinates* $(r, \theta, \phi) = \mathbf{r}$ of Section 2.2 (Fig. 2). Precisely, the new coordinates are given by

$$\delta = r_s(\cos\theta_s)(\phi - \phi_s) = \text{azimuth arc}$$

$$\varepsilon = r_s(\theta - \theta_s) = \text{elevation arc} \qquad (4.1)$$

$$\zeta = r - r_s = \text{relative range},$$

where (r_s, θ_s, ϕ_s) is the secondary origin. The difference between (r, θ, ϕ) and $(\delta, \varepsilon, \zeta)$ consists in two features: (i) offsets to enable the user to locate the new, i.e. secondary, origin (r_s, θ_s, ϕ_s) at an arbitrary point, and (ii) multiplication by a factor to make δ and ε into suitable *arc lengths*. The new coordinates may be interpreted as three successive displacements that must be carried out, *in the order* $(\delta, \varepsilon, \zeta)$, to get from the secondary origin to the image point P: first along the circle along which ϕ alone varies ($\theta = \text{constant}$, $r = \text{constant}$), then along the great circle along which θ alone varies, and finally along a radial line. These operations are shown in Figure 2 (and in Fig. 10). The grid points are generated as the points of intersection of one of a family of surfaces $r = \text{constant}$ simultaneously with one of a family of surfaces $\theta = \text{constant}$ and one of a family of surfaces $\phi = \text{constant}$. (Figure 10 shows one such three-way intersection.) Each planar plot will be of one such surface.

4.2.2.2 Viewwindows

We digress to again discuss a general viewgrid. Such a viewgrid is described by (i) the *viewgrid centre* $(\delta_c, \varepsilon_c, \zeta_c)$, which need not coincide with the secondary origin [the latter being at $(\delta, \varepsilon, \zeta) = (0, 0, 0)$], and (ii) the *viewwindow* $(\Delta\delta, \Delta\varepsilon, \Delta\zeta)$, which gives the dimensions of the viewgrid. The significance of these two vectors for linear graphs is that, for instance, the first of the three graphs extends along the line $(\varepsilon = \text{constant}, \zeta = \text{constant})$ through the viewgrid centre from $\delta = \delta_c - \Delta\delta/2$ to $\delta = \delta_c + \Delta\delta/2$.
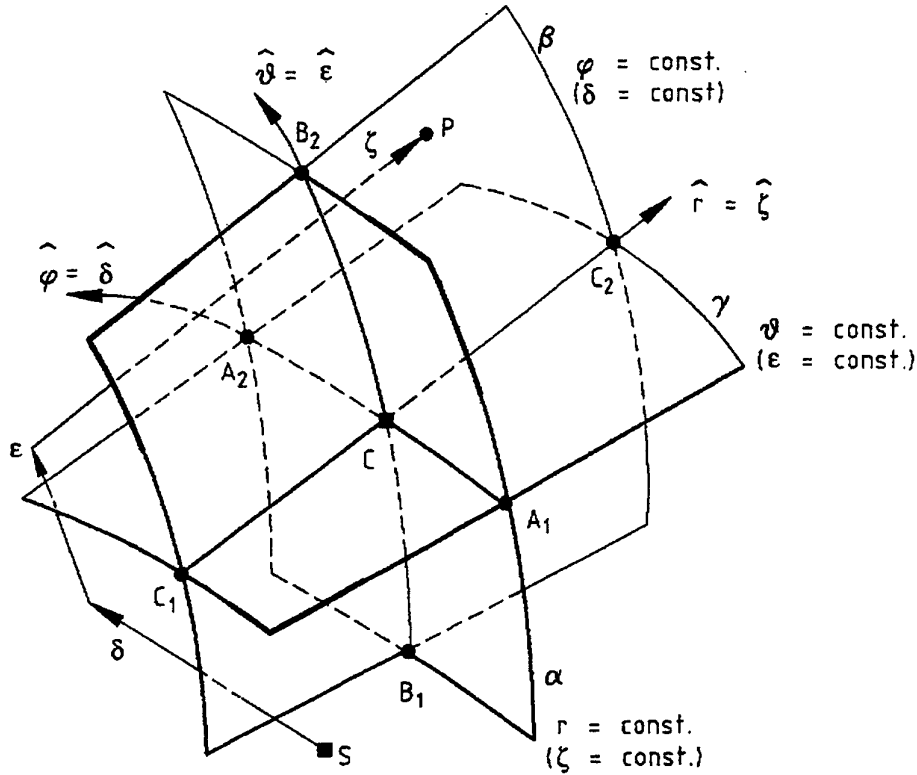
Figure 10: *The viewwindows for the unrotated radially-curved case. For linear plots the viewwindows are A_1CA_2, B_1CB_2 and C_1CC_2. For planar plots the viewwindows are the three intersecting curved 'rectangles'. The path from S to P is the same as in Figure 2. The significance of the secondary origin S is that, on the graphs, the coordinate labels are proportional, not to ϕ, θ and r, but to $\phi - \phi_s$, $\theta - \theta_s$ and $r - r_s$, where the subscript s refers to the point S. Indeed the labels themselves are numerically equal to δ, ε and ζ.*

In the unrotated, radially-curved case, the above features are illustrated by Figure 10; there the linear viewwindows $\Delta\delta$, $\Delta\varepsilon$ and $\Delta\zeta$ are A_1CA_2, B_1CB_2 and C_1CC_2. In that case, from Equation (4.1), the viewwindow $\left(\Delta\delta, \Delta\varepsilon, \Delta\zeta\right)$ in δ, ε and ζ is related to the corresponding viewwindow $\left(\Delta r, \Delta\theta, \Delta\phi\right)$ in r, θ and ϕ by

$$\Delta\delta = r_s\left(\cos\theta_s\right)\Delta\phi$$

$$\Delta\varepsilon = r_s\Delta\theta \qquad (4.2)$$

$$\Delta\zeta = \Delta r$$

29

Most often the viewgrid centre and the secondary origin are chosen to coincide—see Section 4.2.2.3.

We digress a second time to discuss general viewgrids. In *planar* graphs, the viewgrid points are distributed over each of three surfaces ζ = constant, δ = constant and ε = constant. Each of the three *planar* viewwindows is formed from two linear viewwindows as indicated in the diagram by the formation of 'rectangles'.

4.2.2.3 Untranslated and Translated Options

In the program, when the unrotated radially curved grid type is used, there is a default— the 'untranslated' case —in which the viewgrid centre coincides with the secondary origin. (Then in Figure 10, S and C coincide.)

However, the user may choose to 'translate' the viewgrid centre away from the secondary origin. This feature is useful when making detailed plots of parts of the tail of a point spread function. In that case, it is most convenient to locate the secondary origin at the target, which is constant from graph to graph, while the viewgrid centre varies between graphs. The advantage is that a common coordinate system for labelling then applies across all the graphs.

4.2.3 Other Viewgrids

We now discuss the '*rotated*' radially curved case, which is available only for linear graphs. In it, following the transformation (4.1), a rotation in $\delta\varepsilon$ space is performed, through an angle α chosen by the user. In ordinary 3-D space, the effect is to perform something that approximates a rotation: the radial line through the secondary origin $\left(r_s, \theta_s, \phi_s\right)$ remains fixed ('axis' of rotation) and all radial coordinates r remain unchanged. Consider for example a target at broadside; then the 'rotated' facility is useful when one wishes to plot the amplitude along a direction, on the sphere r = constant, that is oblique to the principal axes of the square receiving array.

There are also 'flat' viewgrid types. In these, in general there is a translation of the *xyz* axes followed by a true rotation, to produce the new or $\delta\varepsilon\zeta$ axes. The viewgrid is attached to the latter axes (Fig. 10, but with flat surfaces).

4.2.4 Results

The radially curved grids (unrotated and rotated) have been found to be far more useful than the flat types. In particular, in a point spread function, the beam pattern along the spherical surface through the target point is found to follow approximately the usual far-field beam pattern for the same array, at least for the near sidelobes. And the pattern along the radial line through the target closely reflects the ensonifying waveform. No doubt these features of radially curved grids are due to the facts that first, the spherical surfaces follow the ensonifying wavefront, and second, the surfaces

also follow approximately the backpropagated wavefront from each receiving element when the aperture is small.

4.3 Normalisation

4.3.1 Normalisation in the Main Program: Quasinormalised Amplitudes

By 'normalisation' we mean the choice of the premultiplying 'constants' in the expression for the image amplitude. We first discuss normalisation within the main program POINTSPR. The set of choices made in this Section 4.3.1 yields what we shall call *'quasinormalised'* amplitudes. In the course of the discussion, some theoretical properties of the simulation model are derived.

First, in Equations (2.1) and (2.2), the program chooses

$$K_1 = K_2 = 1. \tag{4.3}$$

In the second group of normalisations, the element coupling strengths in Equations (3.2), (3.5) and (C.1) respectively are chosen as

$$K_3 = 1, \qquad K_5 = 1/(M'd)^2, \qquad K_6 = 1/(M')^2. \tag{4.4}$$

The last two choices in (4.4) are motivated *initially* by Equations (3.3), (3.11) and (C.7) and the comments following the last two of these equations. However we now present a superior motivation, which applies to the more general case $b \neq 0$, and is applicable also to the case of a correlated signal. Use is made of image-forming concepts discussed in Sections 5.1 and 5.2, and of expressions for the image amplitude along a radial line (Section 5.1.3).

To express the result that motivates (4.4), let us define the *'coincident'* image amplitude to mean the image amplitude at the target point r_1, it being implied that there is only a single target (labelled $j = 1$). We denote by $\Pi_0(r_1)$ the coincident *in-phase* amplitude (ignoring for brevity the corresponding total amplitude). For pointlike elements, under a set of three conditions (to be given), Π_0 is given by

$$\Pi_0(r_1) = K_3 a_1 \frac{\sigma(r_1, 0)}{r_1^2} \sum_n w_n \tag{4.5}$$

for both uncorrelated and correlated signals (the normalisation 4.3 being imposed). Under the same conditions, the results for solid and tilelike elements are identical to (4.5), but with K_3 replaced by $K_5(M'd)^2$ and $K_6(M')^2$ respectively. The motivation for (4.4) is now obvious.

The three conditions are as follows. First, the target is at broadside (r_1 on the z axis). Second, the target is in the *'relaxed* far field' of the array, that is

$$r_1 \gg Md, \tag{4.6}$$

that is, its range is much greater than the aperture of the array. Third, it is required that

$$r_1 \gg (Md)(M'd)/2\lambda_c ,\qquad(4.7)$$

where the right-hand side is the geometric mean of two near-field-far-field transition distances: the transition distance for the array and that for an element.[19]

The result (4.5), and its equivalents for extended elements, under the three conditions stated, follow readily from Section 5.1.3 (especially Eqn 5.14).[20]

The significance of the normalisation (4.4) is that if the elements, initially of one type, are replaced by elements of another type, without changing the placement or weighting of the elements, the coincident amplitude remains unchanged, subject to the three conditions. Thus the three types of element are given equal 'weight' in determining how large the quasinormalised amplitudes are.[21]

A final point regarding the second group of normalisations (4.4) follows because the result (4.5) continues to hold for correlated signals. Thus with the current normalisations (mainly $K_1 = 1$ in 4.3), uncorrelated and correlated signals are given equal 'weight.'

The normalisation in the third and final group concerns the weighting of the array elements. In the program, the normalisation of the set of weights is given by

$$\sum_n w_n = N ,\qquad(4.8)$$

where N is the number of elements. The significance of this choice can be seen by considering the formula (4.5) and its equivalents. Thus all apodisations (all sets of relative weights) receive equal 'weight', in that changing the apodisation leads to no change in the coincident amplitude, under the three conditions.[22]

4.3.2 Normalisation for Plotting

Further normalisation occurs in the plotting routines PLINE and PLPLANE. That is, further premultiplying constants are chosen, to be applied immediately prior to plotting. A rough description of the default is that each scene is normalised so that the

---

[19] Actually the *pointlike* and *tilelike* elements do not require conditions as strict as these three in order for (4.5) (or its tilelike equivalent) to hold.

[20] In that section, the factor N must everywhere be replaced by $\sum_n w_n$, since the normalisation (4.8) has not yet been imposed.

[21] Note however the following corollary: *subelements* in a tilelike element are *not* given the same weight as would be given to the pointlike elements in the array formed by converting every subelement in every tilelike element into a pointlike element. (This corollary makes a different point from the similarly-worded statement made below Eqn 3.12.)

[22] Actually the same property would be possessed by any normalisation that put the sum of weights in (4.8) equal to some constant, i.e. some value independent of the apodisation. The particular normalisation (4.8) ensures that increasing the number of elements increases the coincident image amplitude proportionally, which is reasonable.

amplitude of the 3-D image is equal to unity at the first target point (first in the list supplied by the user).

4.4 Spatial 'Averaging'

Spatial 'averages' of the image intensity are often useful; the inverted commas are added to include similar processes, in particular the taking of the maximum.

4.4.1 Averaging Within the Curve or Surface

In the linear plotting routine (PLINE), the option exists to calculate the average (arithmetic mean of the image intensities, not the amplitudes or decibel values) along the curve being plotted, optionally excluding some interval. The result is called a 'sidelobe average.' The exclusion region may be freely chosen, but normally it would include the main lobe and some of the nearer sidelobes.

This averaging feature is particularly useful in the case of random arrays, whose properties must be described by averages (Section 3.6). It is the average over the distant sidelobes that is of interest, because this determines the 'clutter,' which is analogous to the fog level in a photograph.

In the planar plotting routine (PLPLANE), a similar option exists. In this case the exclusion region is rectangular; at present the exclusion centre must coincide with the viewgrid centre.

4.4.2 'Averaging' over Volumes

D.E. Robinson [private communication] has been pointed out a feature that preferably would be incorporated into the simulation program, but at present is not. The feature has been usefully exploited by CSIRO Department of Telecommunications and Industrial Physics. Consider the image of a point target. Because of the randomness of the array, isolated peaks in the clutter occur, and *these are not necessarily on the principal planes of the image.* Also, for multiple targets these peaks need not be near any of the targets. CSIRO performed calculations for image points throughout a volume. For presentation they used the maximum intensity along the ray in the direction normal to the plane of the display. Robinson reports that sidelobe values using this method are considerably greater than those obtained with no maximisation process (calculating for principal planes only), and that the values better match reality.

5. Image-Forming

5.1 Image Before Demodulation

5.1.1 Derivation of Image Amplitude Function

In this section we develop, along with a supporting argument, the 'delay-and-add' image-forming formula for the general near-field, broadband, 3-D-image situation. Although this formula (Eqn 5.4 below, with σ ignored), is known, it is seldom seen in the sonar literature; it is described in words in Knudsen [1989]. Image-forming in the *far* field is described in the five references on arrays listed at the start of Section 3; image-forming in the second-order approximation and the Fresnel approximation also are described in some of these references.

To see the force of the supporting argument for delay-and-add image-forming, it is best to think of a waveform $\xi(t)$ whose envelope has three properties: (i) it is symmetric, (ii) the envelope falls off monotonically and smoothly as $|t|$ increases, for example a gaussian envelope, and (iii) the width (standard deviation) of the envelope contains just a few cycles of the carrier wave. The results do however apply to more general waveforms. It is best also to think of pointlike elements. However, down to and including Equation (5.4), the results apply also to extended elements.

Towards forming our 'image' amplitude at $\mathbf{r}$ (the image point—see Fig. 1), let us for now restrict attention to the uncorrelated, nonquantised case. We form the weighted and delayed sum

$$\Pi(\mathbf{r}, t) = \sum_n w_n \, \sigma(\mathbf{r}, n) E_n[t + \tau(\mathbf{r}, n)]. \tag{5.1}$$

Here w_n (real) is the *weight* given to the nth element (Section 3.4) and $\sigma(\mathbf{r}, n)$ is either unity (or a constant) or a spherical spreading correction factor (see Section 5.3.1); while $\tau(\mathbf{r}, n)$ is the functional relationship defined by Equation (3.4), so that

$$\tau(\mathbf{r}, n) = (r + |\mathbf{r} - \mathbf{R}_n|)/c \tag{5.2}$$

is the round-trip time that would be taken by a pulse travelling from the projector to the nth element via the image point $\mathbf{r}$. For future use we introduce also the round-trip distance

$$\Lambda(\mathbf{r}, n) = c\tau(\mathbf{r}, n) = r + |\mathbf{r} - \mathbf{R}_n| = r + (r^2 - 2\mathbf{r} \cdot \mathbf{R}_n + R_n^2)^{1/2}. \tag{5.3}$$

For a single scatterer at $\mathbf{r}$, the envelope of the function $\Pi(\mathbf{r}, t)$ will be peaked at, or very near, $t = 0$. To check this claim, note that at $t = 0$, not only are the values of E_n near their maximum, but, from Equation (3.3), the values of E_n for the various elements combine *in phase* with each other, it being assumed that w_n and $\sigma(\mathbf{r}, n)$ are positive. We therefore assume that the system takes the value of $\Pi(\mathbf{r}, t)$ at $t = 0$ to be the *in-phase (or modulated) image amplitude function* $\Pi(\mathbf{r})$ at $\mathbf{r}$; thus

$$\Pi(\mathbf{r}) = \Pi(\mathbf{r}, 0) = \sum_n w_n \, \sigma(\mathbf{r}, n) E_n[\tau(\mathbf{r}, n)]. \tag{5.4}$$

Equation (5.4) will be called the *image-forming equation*. The function $\Pi(\mathbf{r})$ is a modulated or 'radiofrequency' signal: it oscillates rapidly with range. The final result, or *total image intensity function*, the intensity of the image at $\mathbf{r}$, is the square of the envelope of $\Pi(\mathbf{r})$; it is obtained by processing $\Pi(\mathbf{r})$ in one of two ways, to be discussed in Section 5.2.

We shall argue that, *within the context of the simulation model*, the expression (5.4) yields an acceptable 'image intensity', i.e. that the image resembles the target. Within the model, using Equations (5.4), (3.3) and (3.4), it follows that, for pointlike elements,

$$
\Pi(\mathbf{r}) = K_2 K_3 \sum_j \frac{a_j}{r_j} \sum_n w_n \, \sigma(\mathbf{r}, n) \frac{1}{|\mathbf{r}_j - \mathbf{R}_n|} \xi\left[\left(r + |\mathbf{r} - \mathbf{R}_n| - r_j - |\mathbf{r}_j - \mathbf{R}_n|\right)/c\right]
$$

$$
= K_2 K_3 \sum_j \frac{a_j}{r_j} \sum_n w_n \, \sigma(\mathbf{r}, n) \frac{1}{|\mathbf{r}_j - \mathbf{R}_n|} \xi\left[\tau(\mathbf{r}, n) - \tau(\mathbf{r}_j, n)\right].
$$

(5.5)

We now check that $\Pi(\mathbf{r})$ has certain properties that we expect of an image amplitude function in the case of a single point scatterer j. First note from (5.5) that, as $\mathbf{r}$ moves away from $\mathbf{r}_j$ in the range direction, $\Pi(\mathbf{r})$ soon decreases greatly in the case of a short pulse $\xi(\cdot)$, the decrease occurring in a distance of order c times the pulse duration. Second, as $\mathbf{r}$ moves away from $\mathbf{r}_j$ in the transverse directions (r =constant), eventually there is interference between the contributions from the ξs of the different elements n; that is, these contributions are no longer in phase and they tend to cancel each other. A calculation shows that this cancellation occurs in an angular displacement of order λ_c/(array length), the same formula as in the well-known far-field result. In all directions therefore, $\Pi(\mathbf{r})$ is fairly sharply peaked at $\mathbf{r}_j$. This is in accordance with what one would demand of a point spread function in a good imaging system.

The above image-forming argument still goes through for more general imaging systems in which both the elements and the projector are distributed arbitrarily in 3-D space (instead of all lying in the one plane). The only change required in the formulae is that, in place of (5.2), the round-trip time to be used becomes

$$
\tau(\mathbf{r}, n) = \left(|\mathbf{r} - \mathbf{R}_p| + |\mathbf{r} - \mathbf{R}_n|\right)/c,
$$

(5.6)

where $\mathbf{R}_p$ is the location of the projector.

The above argument for the image-forming equation (5.4) *is based on a model* of the system—including the medium and the target. When one takes the step of applying (5.4) to a real system, the assumption in the model that is believed to be most bothersome is that the target is a collection of isotropic point scatterers with each bundle of energy being scattered once only. Obviously one is interested in the conditions under which the application of this assumption to real systems is justified.

Solving this problem by analytical methods is a difficult task.[23] Hence any study of the conditions must proceed either by (i) testing the method's success in operational imaging, or (ii) testing it in a series of simulations in which surfaces or other targets are accurately modelled. In regard to (i), the image-forming method (5.4) has on numerous occasions *been used in practice* with considerable success — in medical ultrasound and in the experimental AMI system. In particular, experiments performed by Thomson Marconi Sonar and CSIRO with our assistance have found that *rough* surfaces are imaged well. It is however believed that the method is not the last word. In particular, when specular reflections occur, other simple methods may complement the image-forming equation [D. McLean, private communication].

The image intensity and amplitude as defined above are not quasinormalised (and certainly not normalised) — see Section 4.3.

5.1.2 Image Amplitude as a Function of Angular Displacement

Computer simulations and analytic calculations based on the model's assumptions allow us to make more precise statements on the variation of the point spread function with position. In this section we discuss the variation with angular displacement. The first result concerns a toneburst $\xi(t)$ containing many cycles. Consider the point spread function on the sphere $r = $ constant through the point target, located any radial distance (whether in the near or the far field). Then — for a considerable number of sidelobes from the peak — the interference between the contributions of the various elements leads to virtually the same sidelobe pattern as in the familiar one obtained in the *far field*.[24]

The second result from the simulations is that the degree of this agreement with the far-field pattern, described in the first result, extends to many more sidelobes in the case of a target on or near the broadside axis than for targets elsewhere.

Third, similar simulations have been done with broadband, not just narrowband, signals $\xi(t)$. We have used correlated chirps; but it is believed that the result to be stated holds for a very general class of broadband signals. The result is that the same similarity to a monofrequency, far-field beam pattern again applies — the latter pattern being taken at the central frequency. However the similarity is maintained only over a rather small number of sidelobes. For example, if the bandwidth is ¼ of the central

---

[23] However, it is believed that a *smooth* surface can be so modelled provided that the principal curvatures are everywhere convex or zero and that the radius of curvature is very large compared to a wavelength. It appears that the scattering points in the model must be quite closely spaced. All this having been said, smooth surfaces still pose a problem for acoustic imaging because of specular reflections.

[24] In turn the latter pattern is the same as obtained in *passive* detection of a cw source in the far field. This happens because, in the far-field approximation, the path lengths for an active and a passive system associated with a target point r differ by an amount r, the paths being $2r + \varepsilon$ and $r + \varepsilon$ respectively, where usually $|\varepsilon| \ll r$. This constant difference r cancels out in the image-forming process, so that the angular beam patterns are the same.

frequency, a rough similarity extends out to about the 4th node of the sidelobe pattern. We may express this condition of validity in symbols. Let α denote the angular displacement of the point target from broadside, and β denote the angular separation of the image point **r** from the target point $\mathbf{r}_1$. Let L be the side or diameter of the array. Then the condition for the familiar far-field expression to be really good is that

$$\beta << \frac{f_c}{B}\frac{\lambda_c}{L\cos\alpha}; \quad \text{that is,} \quad \beta << \frac{c}{BL\cos\alpha}.$$

The basic ideas in the first and third results above have been reported by Smith *et al.* [1991].

5.1.3 Image Amplitude as a Function of Range

Concerning the variation with range, computer simulations, and also the theory to be given below, yield the following result. Consider the case of a single point target located at $\mathbf{r}_1$. As **r** moves *along a radial line* away from $\mathbf{r}_1$, *the in-phase amplitude function* $\Pi(\mathbf{r})$ *is an accurate* (though not exact) *reflection of the shape of the waveform* $\xi(t)$, with the scaling factor between range and time given by $2r = ct$. In the correlated case of Section 6, replace $\xi(t)$ by $Y(t)$ in this result. Similarly the *total* amplitude function is an accurate reflection of the shape of the *envelope* of $\xi(t)$.

5.1.3.1 *Simple, Less Rigorous Treatment*

The connection just mentioned, between $\Pi(\mathbf{r})$ and $\xi(t)$, will be explained first by an intuitive, less rigorous argument. We consider just pointlike elements. Then Equation (5.5) gives us a connection between r and t. The connection is between t, defined as the argument of ξ in (5.5), and **r**, which occurs in the first term of that argument, when **r** is parallel to $\mathbf{r}_1$. The relationship is

$$2(r - r_1) = ct. \tag{5.7}$$

For, first, when $\mathbf{r} = \mathbf{r}_1$, t is precisely zero. And second, when r increases by Δr, t increases by Δt, where

$$c\,\Delta t = 2\Delta r; \tag{5.8}$$

this follows from (5.2), the approximation being made that $\mathbf{R}_n = 0$. The factor 2 comes essentially from the two-way path. Note that there is no requirement for the target $\mathbf{r}_1$ to be near broadside.

Some results immediately follow. Just as one cycle of the carrier wave *in time* is $1/f_c$ (for $b = 0$, and on average for $b \neq 0$), so *one cycle in range* — that is, one cycle of the modulated image amplitude Π — is $c/2$ times this, or $c/2f_c$. By putting $t = \pm T/2$, we obtain for the ranges at which Π is cut off the values

$$r = r_1 \pm cT/4 \quad \text{or} \quad r = r_1 \pm cT/2. \tag{5.9}$$

The first of these two results applies for an uncorrelated signal. For a correlated signal, Y is cut off at $t = \pm T$ and so the second result is obtained. Some related results are

given in Appendix E, where we develop the concepts of the 'umbra', the 'penumbra' and the 'fully ensonified region' in the 3-D space of the image point, and obtain the boundaries of these regions.

5.1.3.2 More Rigorous Treatment

The more rigorous treatment of radial dependence is given in Appendix F. Subject to the conditions of validity to be given below, it is found that the in-phase image amplitude, along a radial line through the single point target at $\mathbf{r}_1$, is

$$\Pi(\mathbf{r}) = K_3 Na_1 \frac{\sigma(\mathbf{r}, 0)}{r_1^2} \xi\big[2(r - r_1)/c\big] \qquad (5.10)$$

for the case of pointlike elements. The corresponding result for solid elements is:

$$\Pi(\mathbf{r}) = K_5 (M'd)^2 Na_1 \frac{\sigma(\mathbf{r}, 0)}{r_1^2} \xi\big[2(r - r_1)/c\big]. \qquad (5.11)$$

In these results, $\sigma(\mathbf{r}, 0)$ means the spherical spreading correction factor $\sigma(\mathbf{r}, n)$ evaluated for a fictitious element located at the origin. Thus from Section 5.3.1,

$$\sigma(\mathbf{r}, 0) = r_m^2 \quad \text{or} \quad r^2 \qquad (5.12)$$

for the first option (Eqn 5.23) and the second option (Eqn 5.24) respectively. In the results (5.10) and (5.11), all the normalisation conditions of Section 4.3.1 are imposed except (4.4).

For *correlated* signals, the two results (5.10) and (5.11) again apply, with the sole change that ξ must be replaced by its autocorrelation function Y. (This result follows immediately from the discussion to be given in Sections 6.1, 6.2 and 6.6.) For example, the result (5.10) for pointlike elements becomes

$$\Pi(\mathbf{r}) = K_3 Na_1 \frac{\sigma(\mathbf{r}, 0)}{r_1^2} Y\big[2(r - r_1)/c\big]. \qquad (5.13)$$

The results (5.10) and (5.11) make clear that the shape of Π along the radial line is the same as that of ξ (or Y), except for the possible effects of the factor $\sigma(\mathbf{r}, 0)$. In the first σ option — the one normally used — σ is constant and so does not affect the shape of Π.[25]

The similarity between Equations (5.10) and (5.11) is obvious and is discussed elsewhere (Eqn 4.4).

The above radial-line expressions for $\Pi(\mathbf{r})$ are subject to conditions of validity, given in Appendix F.

---

[25] In the second σ option, σ will introduce a gradual modulation into the shape. Even that modulation will be a small effect if the results are applied within an interval such that $|r - r_1| \ll r_1$, as normally they would. Normally that interval is large enough to encompass virtually all the range sidelobe structure.

For the *quadrature* image, in (5.10) and (5.11) replace Π and ξ (or Y) by Π_q and ξ_q (or Y_q) respectively. For the total image amplitude, replace them by Π_t and the analytic signal ξ_a (or Y_a) (and take the absolute value of the right-hand side).

5.1.3.3 Discussion

The corresponding *coincident* values of the image amplitude follow by putting $r = r_1$ in the formulae of Section 5.1.3.2. Thus from (5.10), for the uncorrelated, pointlike case,

$$\Pi_0(\mathbf{r}_1) = K_3 N a_1 \frac{\sigma(\mathbf{r}_1, 0)}{r_1^2} \xi(0), \tag{5.14}$$

where the subscript 0 on Π means 'coincident'. The analogous equation holds for the quadrature part. But $\xi(0) = K_1 = 1$ (Eqns 2.1 and 4.3), and $\xi_q(0) = 0$ (from Eqn 5.22, or more generally from 5.19 and the even function 2.1). Hence the coincident total image amplitude Π_{t0} is

$$\Pi_{t0}(\mathbf{r}_1) = K_3 N |a_1| \frac{\sigma(\mathbf{r}_1, 0)}{r_1^2}. \tag{5.15}$$

Extended elements lead to (5.15) but with K_3 trivially replaced (Section F.3). For correlated signals, Y behaves like ξ in that $Y(0) = 1$ and $Y_q(0) = 0$ (Eqns 6.5 and 6.8). Hence correlated signals lead to (5.15) (with K_3 replaced if necessary) without further change.

A design under consideration for an AMI sonar has $Md = 0.45$ m; the sonar is to be used at ranges r from 1 m to perhaps 5 m. Clearly for $r = 1$ m the condition (F.2) is not satisfied—an unfortunate result. However for that case, simulations (in particular, results to be shown in Figure 18) show that the expression (5.10) actually remains a good approximation—for the first few range sidelobes. And the breakdown at more distant sidelobes is instructive, for it coincides with the violation of Equation (F.5).

5.2 Demodulation

5.2.1 The Need for Demodulation

So far, only the modulated image amplitude, or in-phase part of the image amplitude (Eqn 5.4), has been described. But, for a point target, it is found that, if we plot as the 'image intensity' simply the square of the in-phase part, the result differs from that desired—a blurred point, with sidelobes perhaps unavoidable. The image obtained has an additional cosine-squared factor that oscillates rapidly in the range direction. These oscillations are reproducing the carrier wave of the waveform $\xi(t)$ by the mechanism described in Section 5.1.3. The oscillations do not represent a property of the target; indeed they are a distraction from perceiving the target's features. It is

desirable to get rid of them. Thus one needs to determine some envelope of $\Pi(\mathbf{r})$, that is, a total image amplitude function, $\Pi_t(\mathbf{r})$.

This point has long been recognised in medical ultrasound (references in Section 2.3.1), where the modulated image $\Pi(\mathbf{r})$ is commonly called the *radio-frequency (RF) signal*, being thought of as a function of the range r or of a corresponding time $2r/c$. There, two approaches have been used to convert from the modulated to the demodulated signal. These will be discussed in turn.

5.2.2 Peak Detection

The first method of demodulation, the method normally used in medical ultrasound systems, is called *'peak detection'*. In this method, for a given voxel, the RF signal $\Pi(\mathbf{r})$ is calculated at several points $\mathbf{r} = \mathbf{r}_i$ along the part of a radial line that lies within the voxel. This collection of values is then full-wave rectified to yield values $|\Pi(\mathbf{r}_i)|$. The maximum of these values is then taken as the total, or demodulated, image amplitude Π_t for the voxel. One error in this method is that, due to insufficiently fine spatial sampling, in general the maximum of the continuous signal will be missed, and hence the envelope will be underestimated. A countervailing effect occurs when the envelope varies significantly within a voxel, for then the selection of the maximum tends to overestimate the typical or rms envelope value within the voxel.[26]

5.2.3 Demodulation via the Analytic Signal

5.2.3.1 Introduction

The second method is to use the analytic, i.e. complex, signal. This is the preferred method in research work in medical ultrasound. The problem of missing the peak is avoided. As a side-benefit, one only needs to make a calculation at one point in the voxel.

In the following sections we give the theory of the analytic-signal method for the case of uncorrelated signals. The *Hilbert transform* of a function $f(x)$ is defined as

$$\mathcal{H}(f) = \hat{f}(y) = g(y) = \frac{1}{\pi} \mathrm{P} \int_{-\infty}^{\infty} \frac{1}{y-x} f(x)\,dx , \qquad (5.16)$$

---

[26] This second error is only partly remedied by the analytic signal method. In it, the voxel intensity is normally calculated at a single point (centre of voxel), rather than being spatially averaged.

where P denotes the principal value of the integral and the argument (x or y) in the leftmost expression has been suppressed.[27] Then the Hilbert transforms of $\cos x$ and $\sin x$ respectively are $\sin y$ and $-\cos y$. The *inverse Hilbert transform* is given by

$$f(x) = -\frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{1}{x-y} \hat{f}(y) \, dy$$

An excellent discussion of the analytic signal, including the Hilbert transform, is given by Rihaczek [1985]. Good references on the core aspects of signal processing, including the analytic signal, include Bateman and Yates [1988] and Proakis and Manolakis [1992].

Use will also be made of the Fourier transform (FT) operator $\mathcal{F}$ and its inverse, defined by

$$H(f) = \mathcal{F}[h(t)] = \int_{-\infty}^{\infty} h(t) \, e^{-j2\pi f t} \, dt$$

$$h(t) = \mathcal{F}^{-1}[H(f)] = \int_{-\infty}^{\infty} H(f) \, e^{j2\pi f t} \, df,$$

(5.17)

where $j = \sqrt{-1}$. Given a function $h(t)$ and its FT $H(f)$, the FT of the Hilbert transform $\hat{h}(t)$ is given by

$$\mathcal{F}[\hat{h}(t)] = jH(f) \qquad f < 0$$
$$= -jH(f) \qquad f > 0$$

(5.18)

[Bateman and Yates 1988].

5.2.3.2 *Method*
The analytic signal from the projector is

$$\xi_a = \xi + j\xi_q.$$

Here ξ or $\xi(t)$ is the actual or in-phase signal (Eqn 2.1); while

$$\xi_q = \hat{\xi} = \mathcal{H}\xi,$$

(5.19)

the Hilbert transform of ξ, is called the *quadrature* signal. Recall also that $E_n(t)$, the voltage received at the n th element, is given in the model by Equation (3.3) or (3.11).

In the analytic-signal method, the operational system first processes $E_n(t)$ to obtain its Hilbert transform[28] $E_{qn}(t) = \mathcal{H}(E_n)$. The analytic signal received is then

$$E_{an} = E_n + jE_{qn}.$$

---

[27] Some authors insert a minus sign in front of the right hand side; in our nomenclature such an insertion yields the inverse Hilbert transform.

[28] Actually, the operational system performs a *discrete* Hilbert transform, yielding an *approximation* to $E_{qn}(t)$. Note also that systems often calculate the Hilbert transform, not directly via the integral (5.16) or the corresponding sum, but via the Fourier transform and the result (5.18), as discussed later.

Note that $E_{qn}(t)$ is also the signal that would have actually been received had the projected signal been $\xi_q(t)$; this follows from Equations (3.3) and (B.1) and the linearity of the Hilbert transform operator.

Second, the system calculates *the 'quadrature image amplitude function'* $\Pi_q(\mathbf{r})$, *by image-forming in precise analogy to (5.4); thus*

$$\Pi_q(\mathbf{r}) = \sum_n w_n \, \sigma(\mathbf{r}, n) \, E_{qn}\big[\tau(\mathbf{r}, n)\big].$$ (5.20)

The *total image amplitude* function is then

$$\Pi_t(\mathbf{r}) = \sqrt{\Pi^2(\mathbf{r}) + \Pi_q^2(\mathbf{r})} \ ,$$ (5.21)

and the image intensity is the square of this. For completeness, we define the analytic image amplitude as $\Pi_a = \Pi + j\Pi_q$, so that $\Pi_t(\mathbf{r}) = \big|\Pi_a(\mathbf{r})\big|$.

5.2.4 Demodulation in the Program: A Further Approximation

The program simulates an operational system in which the analytic-signal method of demodulation is used. However, the program makes an approximation: it replaces the quadrature part $\xi_q(t)$ of the projected signal by

$$\xi_q(t) = K_1 \sin(2\pi f_c t + b t^2) \qquad -\frac{T}{2} < t < \frac{T}{2}$$ (5.22)
$$= 0 \qquad\qquad\qquad \text{otherwise}\,.$$

i.e. the same formula as for $\xi(t)$ (Eqn 2.1) but with the cosine replaced by sine.

The errors in, and conditions of validity of, this approximation are discussed in Appendix G. The main results are as follows. First, consider a single point target, and image points along a radial line through the target. *The error in the image amplitude is small, except when the range r is within* $1/4$ *cycle,* or $c/(8f_c)$ in distance terms, *of the cutoff points.* The effect is that in the program, the graph of image amplitude versus range has two sharp cutoffs; whereas with accurate demodulation these cutoffs are smoothed out to some degree. Along a radial line having an angular displacement from the target, it is expected that the error is small except near the umbra-penumbra boundaries.

Appendix G also considers a numerical example appropriate to the range resolution desired in the AMI system. The size of one voxel in the range direction is compared with the combined size of two range intervals discussed above, each of size $c/(8f_c)$. If the ratio of the first to the second is large, the approximation (5.22) may be said to be good overall. The ratio comes out to be 9.3, and so the approximation is good. In the correlated case, the approximation will turn out to be much better.

5.3 Other Image-forming Theory

5.3.1 Spherical Spreading

The program user may wish to apply some correction factor for the different amounts of weakening, due to spherical spreading, that occur along different acoustic paths. Either of two *spherical spreading correction factors* $\sigma(\mathbf{r}, n)$ may be applied when image-forming.

The 'first' option is to make no such correction; then $\sigma(\mathbf{r}, n)$ is a constant:

$$\sigma(\mathbf{r}, n) = r_m^2, \tag{5.23}$$

where $r_m = 1$ metre. (This constant is selected to give dimensional consistency with Eqn 5.24.)

In the 'second' option, one 'corrects for' spherical spreading, according to the formula

$$\sigma(\mathbf{r}, n) = r |\mathbf{r} - \mathbf{R}_n|. \tag{5.24}$$

As is seen from Equation (5.5), this factor compensates for the two spherical spreading factors in the denominator for any scatterers j at (or near) $\mathbf{r}$. Thus it compensates for two effects. (i) First, it compensates for the variation in the amplitude of the scattered wave over the array. For a single target, without the correction, the contribution of the element *furthest* from the target, to the amplitude near the target, would be less than the contribution of a *typical* element. (ii) Second, this option compensates for the general weakening of the scattered wave as the target point moves further from the array. For two targets of the same target strength but at different ranges, the effect of the correction is to make the image amplitude at the one peak the same as the image amplitude at the other peak.

Simulations have shown that, under the conditions envisaged for the proposed operational system, the use of the second option as opposed to the constant makes little difference to the image of a single point target; that is, the effect (i) is quite small.

One situation where the first and second options lead to noticeably different results for a single point target is the case of a *long* toneburst (not useful for range determination) combined with a small array. For a sufficiently small array, the condition (F.5) on the formula[29] (5.10) continues to be satisfied even when r differs from r_1 by an amount comparable with r_1. Consequently along a radial line, the image amplitudes (5.10) from the two σ options differ by a factor proportional to r^2. If the signal is a sufficiently long toneburst, the rectangular envelope of ξ in $\Pi(\mathbf{r})$ also will extend out to r where $|r - r_1| \sim r_1$. Consequently out to those limits, along the radial line, the envelope Π_t will be constant for the first option, but proportional to r^2 for

---

[29] The derivation is for pointlike elements, but the results hold also for solid elements.

the second. This marked difference, to be called effect (iii), has been confirmed by simulations.[30]

Finally we note a simple result for the coincident amplitude ($\mathbf{r} = \mathbf{r}_1$, Section 4.3.1) that is obtained in the simulation model when the second option is used. Consider pointlike elements, with a single target ($j = 1$), which may be placed *anywhere*. The latter liberty allows an arbitrary direction; more importantly, no far-field condition is required, in contrast to the conditions required for Equation (4.5). Then from Equations (3.3), (5.4) and (5.24), by an argument almost identical to the 'pointlike' argument of Appendix F, with K_1 and K_2 given by (4.3), the coincident amplitude simplifies to

$$\Pi_0(\mathbf{r}_1) = K_3 a_1 \sum_n w_n$$
$$= Na_1,$$

(5.25)

where the step to the second line uses the normalisation equations (4.4) and (4.8). This result also holds in the case of a correlated signal (as discussed above Eqn 5.13).

5.3.2 Time Dependence

We shall call $\Pi(\mathbf{r}, t)$ defined by (5.1) the *time-dependent* image amplitude function. $\Pi(\mathbf{r}, t)$ may be thought of as (proportional to) an estimate of the voltage at the projector at time t, made from the voltages at the receiver elements on the assumption that all the acoustic signals travelled via just the one scattering centre located at the image point. $\Pi(\mathbf{r}, t)$ may be useful in future theoretical developments. Only a brief discussion will be given here.

First, we expect $\Pi(\mathbf{r}, t)$ to vary with t more or less as $\xi(t)$ does, and on the same time-scale; its peak or centre should occur at or near $t = 0$. Second, recall that the system is assumed to take the image amplitude function $\Pi(\mathbf{r})$ to be the function $\Pi(\mathbf{r}, t)$ evaluated at $t = 0$. An alternative way of implementing this definition in a practical system is to perform some integral of $\Pi(\mathbf{r}, t)$ {or $[\Pi(\mathbf{r}, t)]^2$} over time. This could be done in such a way as to capture (say) 90% of the energy of the pulse, making use of the fact that the waveform $\xi(t)$ is known. Thus an alternative estimate of $\Pi(\mathbf{r}, 0)$ might be obtained that makes use of more data. In the inevitable presence of noise, this alternative method might be superior. However, studies to date have not supported this idea.

---

[30] A corollary is that, for the second option, the peak of Π_t is reached at a distance far beyond r_1. This need not be a cause for concern, since such long tonebursts would never be used when good range resolution is desired.

5.4 Option to Output Voltages

An option exists to output the voltages received at the elements. To keep the output manageable, the voltages are not evaluated at all times. Instead, the user selects a particular image point r; at each element (e.g. the nth) the voltage E_n is evaluated at the time appropriate to calculating the image amplitude at the selected point. From (5.4), this time is $\tau(\mathbf{r}, n)$.

6. Crosscorrelation of the Signals

6.1 General

The technique of pulse compression, based on crosscorrelation, has been discussed in some detail by Rihaczek [1985], Ziomek [1985], Burdic [1991] and Kino [1987]. Pulse compression enables good range resolution to be achieved, at the same time allowing the pulse in the water to have low power (as opposed to low energy). The latter feature in turn allows a pulse with high total energy to be transmitted without driving the medium into nonlinear behaviour.

If the crosscorrelation option is selected, the program emulates, via analytic formulae rather than actually performing a numerical crosscorrelation, a system in which the following signal processing is performed. The system takes the signal $E_n(t)$ received at each of the various elements and crosscorrelates it with the input projector signal $\xi(t)$. For each element, *the system passes on the resulting correlated signal, $E_{rn}(t)$, in place of $E_n(t)$, as the input to the later processing steps* as described in Sections 5.1.1 and 5.2.3.2. Thus Equation (5.4) is replaced by

$$\Pi(\mathbf{r}) = \sum_n w_n \, \sigma(\mathbf{r}, n) E_{rn}\left[\tau(\mathbf{r}, n)\right]. \tag{6.1}$$

It remains to define crosscorrelation. Consider the general case of two *complex* functions f and g. The crosscorrelation of the two is defined as

$$f(\tau) \otimes g(\tau) = \left(f \otimes g\right)(\tau) = \int_{-\infty}^{\infty} f^*(t) g(t + \tau) dt \,,$$

where the asterisk denotes the complex conjugate. When f is a real function (as it is in the present section), the asterisk can be dropped. In Equation (6.1), the function $E_{rn}(t)$, defined as

$$E_{rn}(t) = \frac{1}{K_4} \xi \otimes E_n = \frac{1}{K_4} \int_{-\infty}^{\infty} \xi(t') E_n(t + t') dt' \,, \tag{6.2}$$

is the *in-phase crosscorrelation function* or *in-phase correlated voltage*. In (6.2), the factor

$$K_4 = \int_{-\infty}^{\infty} \left[\xi(t')\right]^2 dt'$$

has been included for convenience.[31] K_4 depends only on the parameters of the waveform ξ and therefore is a constant for most purposes.

The result of this crosscorrelation, for a typical chirp containing very many cycles, is that *the duration of the return signal* from each scatterer—as measured by its 3 dB width for example—*is changed from the full length of the chirp to a much smaller value*. In this sense the chirp is 'compressed' by the crosscorrelation, enabling good range resolution.

In the next few sections, we restrict attention to pointlike elements, postponing extended elements to Section 6.6.

6.2 Theory of Exact Crosscorrelation

In this Section 6.2, we develop the formula giving the image amplitude in terms of the chirped signal's exact autocorrelation function. The approximation that is applied to the autocorrelation function in the program is postponed to Section 6.3.

6.2.1 Autocorrelation Function of the Projected Signal

For pointlike elements, substituting Equation (3.3) into (6.2), we find

$$E_{rn}(t) = K_2 K_3 \sum_j \frac{a_j}{r_j \left| \mathbf{r}_j - \mathbf{R}_n \right|} Y \left[t - \tau \left(\mathbf{r}_j, n \right) \right], \qquad (6.3)$$

where

$$Y(t) = \frac{1}{K_4} \xi \otimes \xi = \frac{1}{K_4} \int_{-\infty}^{\infty} \xi(t') \, \xi(t + t') dt' = \frac{\int_{-\infty}^{\infty} \xi(t') \, \xi(t + t') dt'}{\int_{-\infty}^{\infty} \left[\xi(t') \right]^2 dt'} \qquad (6.4)$$

is the *normalised autocorrelation function* of $\xi(t)$, also called the normalised *in-phase* autocorrelation function (of the analytic signal). Comparing (6.3) with (3.3), we see that $E_{rn}(t)$ is given by *the same expression as $E_n(t)$ but with ξ replaced by Y*. Thus the final or correlated voltage has a value *just as though the input projector signal were $Y(t)$ and no crosscorrelation were performed*.[32]

The numerator of the right-hand side of (6.4) is called the unnormalised autocorrelation function, $Y_u(t)$, of $\xi(t)$. The normalised function has the property that

$$Y(0) = 1. \qquad (6.5)$$

It is readily shown that $Y(t)$ is an even function:

---

[31] The convenience of including K_4 arises from its being the normalisation integral in the expression (6.4) for $Y(t)$ below.

[32] Here it is assumed that the medium remains linear under these circumstances.

46

$$Y(-t) = Y(t).\tag{6.6}$$

The results, (6.5) and (6.6), hold regardless of whether $\xi(t)$ has the form (2.1); indeed, (6.6) holds whether $\xi(t)$ is an even function or odd or neither. As discussed in Appendix H, the autocorrelation function is related to the *ambiguity function* $\chi(\tau, v)$ of Rihaczek [1985, p. 119], called the auto-ambiguity function $X(\tau, v)$ by Ziomek [1985, p. 190].

6.2.2 Quadrature Part of the Autocorrelation Function

We proceed by defining $Y_q(t)$, the quadrature part of the autocorrelation function, as *the Hilbert transform of* $Y(t)$. To relate this quantity to ξ and ξ_q, consider all the real correlation functions that arise from the analytic signal. Since the latter is $\xi_a = \xi + j\xi_q$, *its* autocorrelation function involves *four* real correlation functions. As shown in Appendix I, only two of these are independent, since we have:

$$K_4 Y = \xi \otimes \xi = \xi_q \otimes \xi_q$$
$$K_4 Y_q = \xi \otimes \xi_q = -\xi_q \otimes \xi. \tag{6.7}$$

From (6.6), $Y_q(t)$ is an odd function of t:

$$Y_q(-t) = -Y_q(t), \tag{6.8}$$

since the Hilbert transform of an even function is odd. Again the last result does not depend on the form or parity of $\xi(.)$.

6.2.3 The Quadrature Part of the Voltage and the Image

6.2.3.1 General

Two ways present themselves as ways of defining *a 'quadrature' crosscorrelation function* or *quadrature correlated voltage* $E_{rqn}(t)$ and hence a 'quadrature' image. The respective definitions are:

$$K_4 E_{rqn}(t) = -\xi_q \otimes E_n = -(\mathcal{H}\xi) \otimes E_n;$$
$$K_4 E_{rqn}(t) = K_4 \mathcal{H}(E_{rn}) = \mathcal{H}(\xi \otimes E_n), \tag{6.9}$$

where $\mathcal{H}$ denotes the Hilbert-transform operator defined by (5.16) and the argument t has in most cases been suppressed. In each case the image amplitude is then to be calculated as in the last steps of Section 5.2.3.2, by Equations (5.4), (5.20) and (5.21), except that one uses voltages E with the subscript r added. Both definitions are reasonable extensions of previous work: in the first definition, ξ in Equation (6.2) is replaced by $-\xi_q$; in the second, the method is analogous to that used in the

uncorrelated system (Section 5.2.3). We introduce also a third alternative definition, so far unmotivated:

$$K_4 E_{rqn}(t) = \xi \otimes (\mathcal{H} E_n).$$ (6.10)

As Appendix J shows, all three definitions actually define the *same* function $E_{rqn}(t)$.

By taking the Hilbert transform of (6.3), we obtain the result

$$E_{rqn}(t) = K_2 K_3 \sum_j \frac{a_j}{r_j |\mathbf{r}_j - \mathbf{R}_n|} Y_q[t - \tau(\mathbf{r}_j, n)];$$ (6.11)

that is, *a projected signal* $Y_q(t)$ *would yield a received signal* $E_{rqn}(t)$. Note also from the lower left equality in (6.9) that $E_{ran} = E_{rn} + jE_{rqn}$ is the analytic signal associated with the in-phase correlated signal E_{rn} from the n th element.

6.2.3.2 Computation

We now describe a very efficient method for computing the quadrature part $E_{rqn}(t)$ in an operational system; it is intended that this method be used in the AMI operational system. From (6.2) and the top left equality in (6.9), we have

$$K_4 E_{ran}^* = \xi_a \otimes E_n,$$ (6.12)

where the subscript a denotes 'analytic', so that

$$\xi_a = \xi + j\xi_q, \qquad E_{ran} = E_{rn} + jE_{rqn}.$$

With the Fourier transform defined by (5.17), Equation (6.12) can be transformed into the frequency domain with the aid of Equation (J.2), with the result

$$K_4 E_{ran}^*(-f) = \xi_a^* E_n.$$ (6.13)

[Here $E_n(f)$ or E_n denotes the transform of $E_n(t)$.]

The method for computing the quadrature part is based on Equation (6.13) and uses a fast Fourier transform (FFT). $\xi_a^*(f)$, as a function of discrete frequency, is permanently stored. Note that its negative frequency components are zero, so that when the product on the right side of (6.) is computed, its negative components are again zero. (Indeed only half of the multiplications need to be performed.) Then the inverse FFT of the result yields both the in-phase and quadrature components of the dechirped signal E_{ran}, as is seen from the equation preceding (6.13). Essentially, the second component is obtained at no extra cost in operations.

6.3 Autocorrelation Functions: Results and Discussion

This section discusses the program's approximation to the two autocorrelation functions, $Y(t)$ and $Y_q(t)$. Also discussed is the shape of the signal's autocorrelation function, which is related to the shape of the range sidelobes.

6.3.1 Program's Expressions for the Autocorrelation Functions

The expressions for $Y(t)$ and $Y_q(t)$ assumed in the program are obtained by making two approximations.[33] First, in the second line of (6.7), the exact Hilbert transform $\xi_q(t)$ is replaced by

$$\xi_q(t) = K_1 \sin(2\pi f_c t + bt^2) \qquad -\frac{T}{2} < t < \frac{T}{2}$$

$$= 0 \qquad \text{otherwise} \tag{6.14}$$

(compare Eqn 5.16). Second, in calculating the voltage E_m (6.3), instead of using the exact $Y(t)$ (Eqn 6.4) and $Y_q(t)$, the simulation drops a 'sum frequency' term as follows.

We assume that the waveform contains many cycles:

$$f_c T \gg 1, \tag{6.15}$$

and that the relative change in frequency in one cycle is small:

$$|b| \ll f_c^2 . \tag{6.16}$$

These conditions are identical to two of the conditions that arose in the uncorrelated case, namely Equations (G.4) and (G.5). For the in-phase part, under the conditions (6.15) and (6.16), the denominator of (6.4) for $Y(t)$ becomes $K_1^2 T/2$. In the numerator, after substituting (2.1) into (6.4), we rewrite the product of cosines

$$\cos(2\pi f_c t' + bt'^2) \cos\left[2\pi f_c(t + t') + b(t + t')^2\right]$$

as the sum of a 'difference frequency' term and a 'sum frequency' term, using the identity

$$2\cos A \cos B = \cos(B - A) + \cos(B + A) .$$

The approximation consists in dropping the 'sum frequency' term $\cos(B + A)$. The result for the in-phase autocorrelation function is

$$Y(t) = \frac{\sin\left[b|t|(T - |t|)\right]}{bT|t|} \cos(2\pi f_c t) \qquad |t| \le T$$

$$= 0 \qquad \text{otherwise} . \tag{6.17}$$

Similarly for the quadrature autocorrelation function one obtains

$$Y_q(t) = \frac{\sin\left[b|t|(T - |t|)\right]}{bT|t|} \sin(2\pi f_c t) \qquad |t| \le T$$

$$= 0 \qquad \text{otherwise} . \tag{6.18}$$

---

[33]We note that, if one were to write a program to simulate a waveform with a *smooth*, rather than a rectangular, window, the simple replacement of the cosine by sine would be an excellent approximation to the Hilbert transform. It is suspected (from the treatment of signals by Rihaczek [1985]) that also the 'sum frequency' term would be negligible. In that case the errors in the program's approximation to $Y(t)$ and $Y_q(t)$, which are not too bad for the present waveform, would become negligible. Similar remarks apply to the uncorrelated case (Section 5.2.4.1).

In the simulations, Equations (6.17) and (6.18) are used explicitly; no numerical correlation is performed. The two correlated voltages are given by substituting these expressions into (6.3) and (6.11) respectively. The image amplitude is then obtained by summing voltages as at the end of Section 5.2.3.2.

In Appendix K, the results (6.17) and (6.18) are derived in a different way. The appendix also discusses implications of this alternative derivation for the estimates of the program's error.

6.3.2 Range Sidelobes

The range sidelobes that occur for a correlated signal are discussed at this point, because they have a bearing on the conditions and errors of the program's approximation.

Consider a single point target, and points that lie on the radial line through that target. Then, as in Section 5.1.3, the two modulated image amplitude functions are accurate, but not exact, reflections of the shape of $Y(t)$ and $Y_q(t)$ respectively. Here time t is related to range r by Equation (5.7), namely

$$2(r - r_1) = ct,$$

and the associated discussion. As a result, range sidelobes are produced, with the total image amplitude Π_t being proportional to $Y_t = \sqrt{Y^2 + Y_q^2}$.

In the program's approximation, Y and Y_q are given by (6.17) and (6.18); then

$$Y_t(t) = \left| \frac{\sin[bt(T - |t|)]}{bTt} \right| \qquad |t| \leq T \tag{6.19}$$

$$= 0 \qquad\qquad \text{otherwise} .$$

Note that, for the correlated signal, Y is cut off at $t = \pm T$ (not $\pm T/2$). In distance terms, with the target at range r_1, the cutoffs are at $r = r_1 \pm cT/2$. The sidelobes in (6.19) are sinc-like because, for sufficiently small $|t|$, we have

$$Y_t(t) \approx \left| [\sin(bTt)]/(bTt) \right| = \left| \text{sinc}(\pi^{-1}bTt) \right| . \tag{6.20}$$

At larger $|t|$, in (6.19), the 'amplitude' factor $1/|bTt|$ continues to behave as in the sinc function (see middle expression of Eqn 6.20), but the phase in the numerator departs markedly from that of the sinc function.

If, contrary to the program, a *nonrectangular window* were to be used (in the case of a correlated chirp), it would produce an important benefit: *the range sidelobes would be reduced*. Indeed it appears that in general they would be *greatly* reduced, as the following example indicates. First consider a rectangular window. Then from the previous paragraph, for sufficiently small $|t|$, the total amplitude $\Pi_t(r)$, expressed in terms of the corresponding time t given by Equation (5.7), is

$$\Pi_{t\,rel} = \left| [\sin(\pi x)]/(\pi x) \right| \equiv |\text{sinc}(x)| , \tag{6.21}$$

where

$$x = \pi^{-1}|b|Tt = Bt \qquad (6.22)$$

and B is the bandwidth. In (6.21) the value of the amplitude has been expressed relative to the peak value of $\Pi_t(\mathbf{r})$ (which occurs at $t = 0$). Now, as a simple nonrectangular example, consider the case of a waveform $\xi(t)$ having a symmetric *triangular* window of duration T. Then the relative amplitude, again at sufficiently small $|t|$, is found (after considerable calculation) to be

$$\Pi_{trel} = \left\|\left[\sin(\pi x)\right]/\left[\pi x(x^2 - 1)\right]\right\|.$$

By comparing this expression with (6.21) at $x = 2.5, 3.5, \ldots$, it is readily seen that the sidelobes are drastically reduced. However, note that the first sidelobe now occurs at $x \approx 2.5$, not at 1.5. Also note that the main lobe has been widened, extending now out to $x = \pm 2$, not $x = \pm 1$. This is in accord with a well-known phenomenon in beamforming. In both cases, the use of a more *tapered* window with smoother endings leads to a reduction in the sidelobe level but at the expense of a widening of the main lobe.

6.4 Results for Autocorrelation Functions

This section presents a set of typical numerical results obtained for the functions $Y(t)$, Y_q and Y_t. It shows the typical shape of the functions and illustrates the fact that the error in the program's approximation to them is generally small. A detailed discussion of the errors is postponed to Section 6.5.

The computations of the exact Y and Y_q in this and later sections were performed as follows. Scaled variables were introduced as in Section G.1, along with the relative bandwidth

$$\beta = B/f_c = |b|T/(\pi f_c); \qquad (6.23)$$

to obtain the last equality, Equation (6.22) has been used. The true in-phase autocorrelation function $Y(t)$ (Eqn 6.4) was calculated via the FFT. The relationship, expressed in terms of the ordinary FT, is

$$Y(\tau) = K^{-1} \mathcal{F}^{-1}\left[\xi^*(\phi)\xi(\phi)\right], \qquad (6.24)$$

where τ is the scaled time and ϕ the scaled frequency.[34] Here $\xi(\phi)$ is the FT of $\xi(\tau)$ and K is chosen so that $Y(\tau = 0) = 1$. (The result 6.24 follows immediately from Eqns J.2 and 6.4.) Subsequently $Y_q(t)$ was calculated as the Hilbert transform of $Y(t)$ by the method of Section G.1.

Figure 11 plots the true in-phase autocorrelation function $Y = Y(t)$, and the envelope of $\breve{Y}$, where $\breve{Y}$ denotes the program's approximation to Y, for a duration

[34] Due to the 'zero of frequency' used by the FFT in MATLAB, it was necessary to interchange the high-frequency and low-frequency halves of the spectrum before performing the $\mathcal{F}^{-1}$ operation in (6.26).

$f_cT = 40.5$ cycles and a relative bandwidth β of $1/5$. Figure 12 is for $\beta = 1/3.5$, with f_cT slightly changed to 40.25. These values of β are typical for the operational system. A medium rather than a large value of the duration f_cT has been chosen in these examples, for ease of graphical representation. (Each figure has associated values of the processing parameters, m and θ. These are defined in Section G.1.)

AUTOCORRELATION FUNCTION
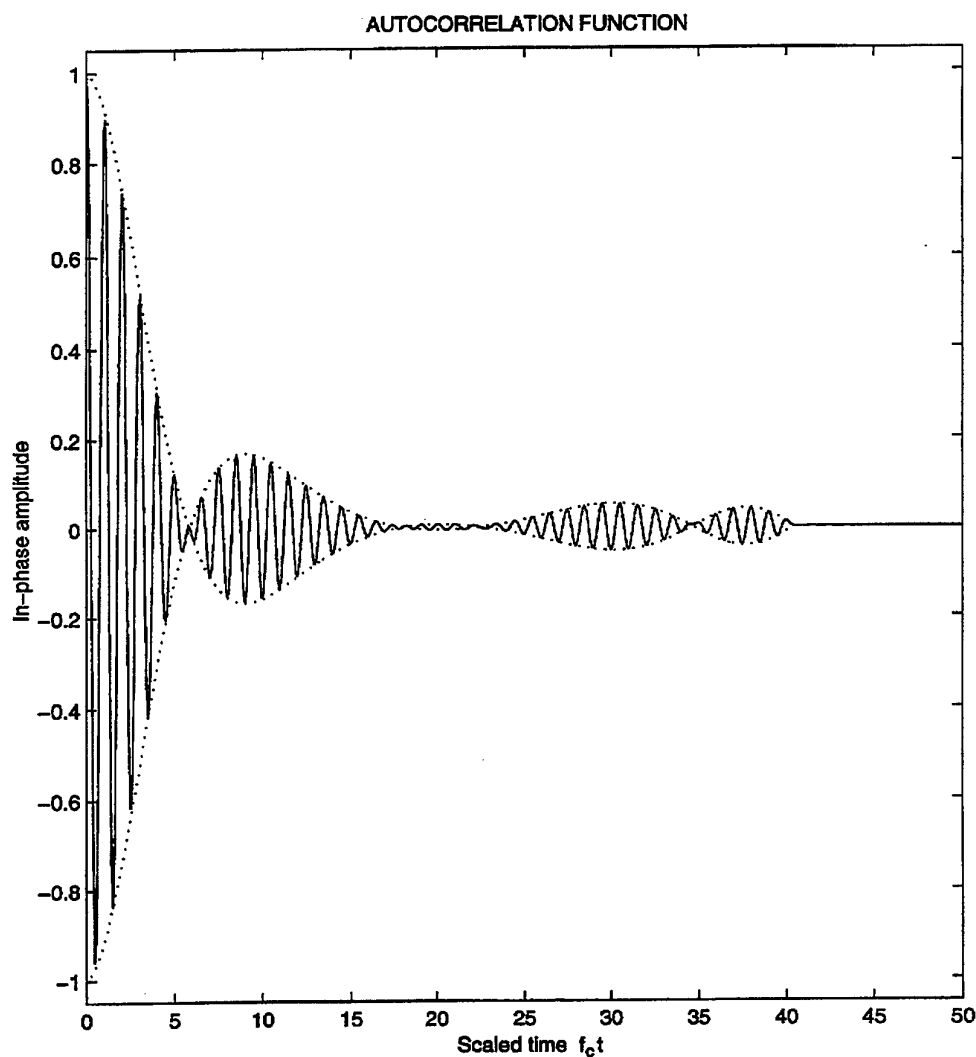
Figure 11: *In-phase autocorrelation function $Y(t)$ (continuous) for $\beta = 1/5$, $f_cT = 40.5$. The envelope of the program's approximation is also shown (dotted). Processing parameters: $m = 4096$, $\theta = 1/8$.*

AUTOCORRELATION FUNCTION
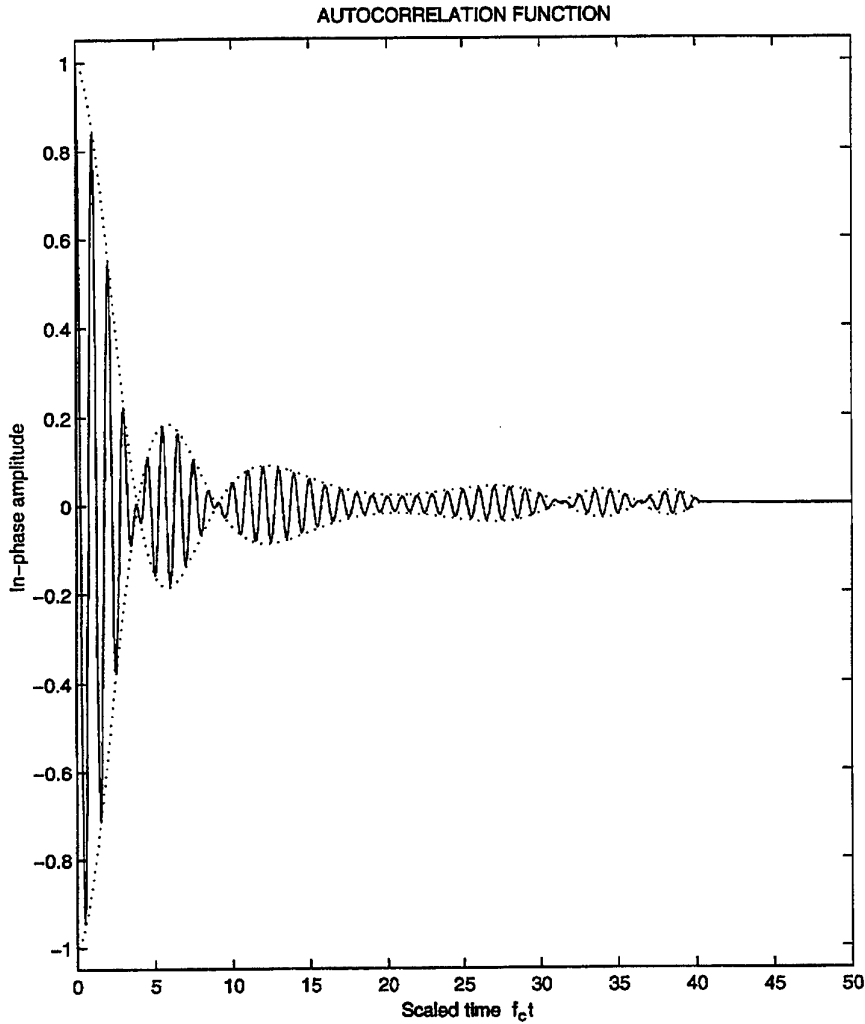
Figure 12: *Same as Figure 11, but for* $\beta = 1/3.5$, $f_c T = 40.25$.

In both cases $\breve{Y}$ (not plotted) is virtually indiscernible from the plot of the true Y, as evidenced by the fact that the respective envelopes are very close together in the plot. The envelope of $\breve{Y}$ resembles a sinc function but with horizontal stretching of the lobes near the middle of the graph. This behaviour is readily explained, given that the envelope equals Equation (6.19). The phase of the sine increases from zero at $t = 0$ to a maximum at $t = T/2$, and back to zero at $t = T$. At the maximum the phase is stationary, giving horizontal stretching in a wide region near $t = T/2$. At $t = T/2$, the

53

envelope can have a low or a non-low value (compared to the peaks of neighbouring lobes), depending on what the maximum value of the phase happens to be; the two figures differ in this respect.

The reader is reminded that these graphs also give the shape of the image amplitude function along a radial line (see Section 6.3.2).

6.5 Autocorrelation Functions: Errors and Conditions

We now turn to errors in, and conditions on, the program's approximation for Y, Y_q and Y_t. We consider the case where Equations (6.15) and (6.16) are satisfied and the 'chirp term' in the phase in (2.1) progresses through many cycles:

$$|b|T^2 \gg 1. \tag{6.25}$$

Note that Equation (6.25) is also the condition for the peak intensity of the last of the range sidelobes before the cutoff to be much smaller than the intensity at the target position or main peak.

6.5.1 An Example of the Error

Figure 13 plots the error, $\delta Y = \breve{Y} - Y$, for the same waveform parameters as in Figure 12 (Section 6.4). Also plotted is the envelope[35] of $\breve{Y}$. Note first that the error, while fluctuating, is, in a root-mean-square (rms) average sense, almost independent of t in $0 \leq t \leq T$. Second, the error is *negligible* compared to the *peak* value, it being borne in mind that Y at its peak is unity (and hence off-scale). Third, the relative errors in the last range sidelobe before cutoff are of interest because it is in this lobe that the relative error — that is, relative to the *local sidelobe peak* — is expected to be worst. Even in this lobe, the relative error turns out to be *fairly small*.

---

[35] At first glance the envelope (dotted) may appear to be aliased, i.e. to overflow from the top to the bottom of the graph, but a closer look shows that no such overflow occurs.
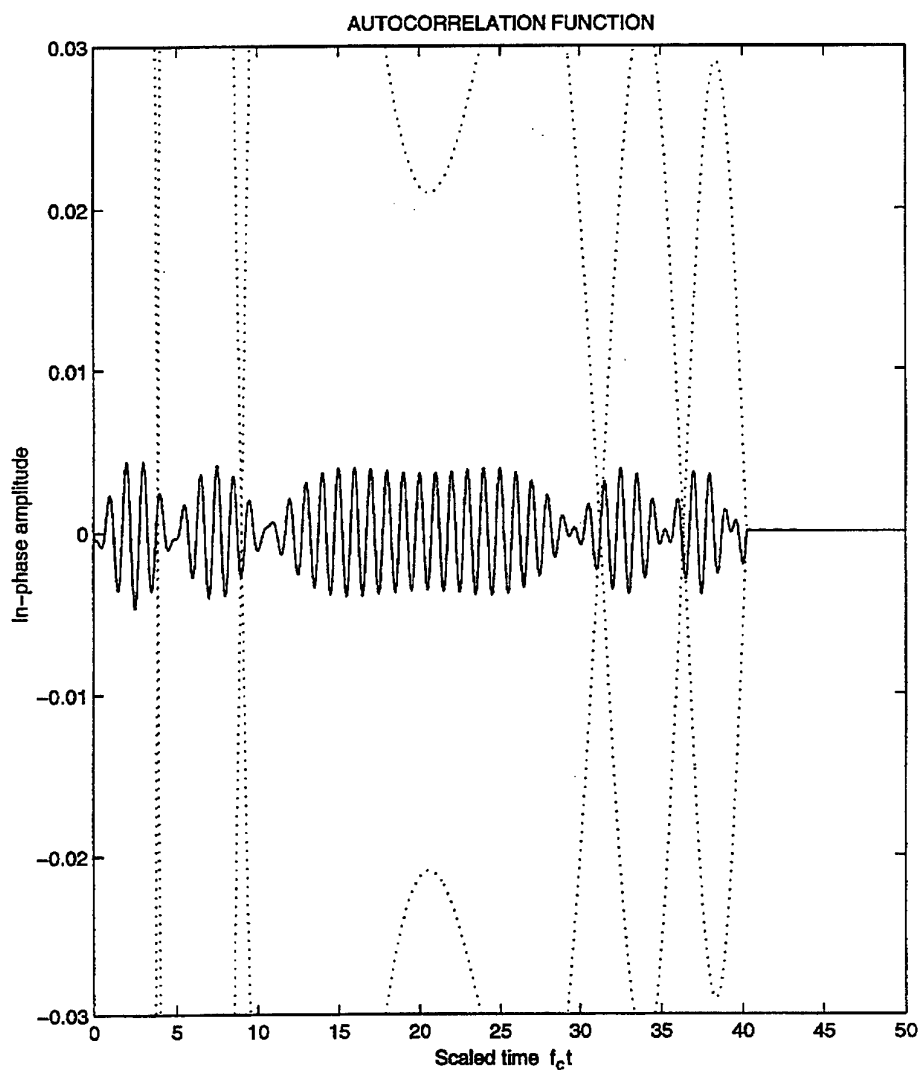
Figure 13: *Error in the in-phase autocorrelation function* $Y(t)$ *for the same pair of waveform parameters as in Figure 12* ($\beta = 1/3.5$, $f_c T = 40.25$). *For comparison, the envelope of the program's approximation to* $Y(t)$ *is shown dotted.* $m = 16\ 384$, $\theta = 1/16$.

AUTOCORRELATION FUNCTION
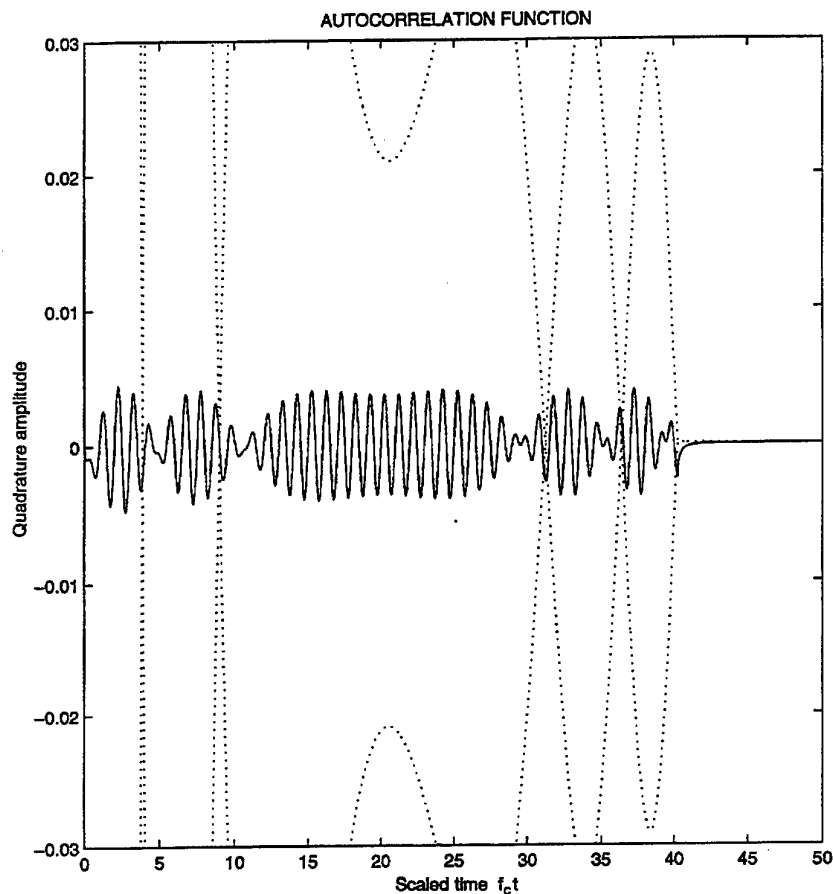
Figure 14: *As for Figure 13, but for the quadrature function* $Y_q(t)$.

For the quadrature amplitude $Y_q(t)$, Figure 14 plots the error $\delta Y_q = \breve{Y}_q - Y_q$, and the envelope of $\breve{Y}_q$, in analogy with Figure 13. Strikingly, the same three properties as in the previous paragraph are found to hold. Moreover, the rms errors in Y and Y_q over $0 \le t \le T$ were calculated and found to be equal:

$$\Delta Y = \Delta Y_q = 0.0023, \tag{6.26}$$

where the notation

$$\Delta Y = \left(\delta Y \right)_{rms}$$

is used for the rms error.[36]

---

[36] Strictly, Equation 6.26 refers to a recalculation with $m = 32\ 768$, $\theta = 1/14$.

6.5.2 Details of Errors and Conditions

The errors and conditions pertaining to the Ys are discussed in detail in Appendix L. The main points are summarised here. First, a calculation is reported, similar to that in Section 6.4, but for a more realistic example of the operating system — more realistic because involving a much longer chirp given by $f_c T = 400$. The computed rms error in the interval $0 \leq t \leq T$ was found to be

$$\Delta Y_t = 5.4 \times 10^{-5}. \tag{6.27}$$

Gratifyingly, this is extremely small (much smaller than the errors in Section 6.4 given by Eqn 6.26).

Second, we saw in Section 6.5.1 that the relative errors in the last range sidelobe before cutoff are of interest. The appendix calculates these errors for a medium-length chirp and also for a long chirp. In both cases the relative errors were found to be fairly small, though not negligible.

In summary so far, the numerical results show that the error due to the program, even in the worst case, is *fairly small*, and for most situations of interest it is *very small*.

Third, Appendix L also attempts to obtain formulae for the errors. An argument of uncertain validity yields for the rms error in the total amplitude

$$\Delta Y_t = \sqrt{2} / (4\pi f_c T), \tag{6.28}$$

and for the error in each of Y and Y_q a value $1/\sqrt{2}$ times this. From the few numerical results to date, these theoretically predicted values, while often a considerable overestimate of the error, appear to give an upper limit on the error.

Finally, Appendix L has implications for the error in the *image intensity function* due to the program's approximations to the Ys. The error has already been calculated along a radial line through the point target [since this dependence is a reflection of $Y_t(t)$] and is found to be small in practical situations. Along other radial lines, we may expect somewhat similar errors. The relative error should be greatest near the umbra-penumbra boundary.

6.6 Solid Elements and Tilelike Elements

The combination of crosscorrelation with solid elements is discussed in Appendix M. The result for the correlated signal output from the element, $E_m(t)$, is given by Equations (M.19), (M.20) and (M.16); $E_{rqn}(t)$ is also given. The same derivation also yields the results for tilelike elements.

For solid elements, the conditions of validity for these results are of two types; and corresponding to each condition of validity there is an error. The first-type errors are the same as for pointlike elements, and are given by the errors in Y and Y_q discussed in Section 6.5. The second-type condition of validity consists simply of the far-field condition, given as the display equation below (3.7). Tilelike elements require the same

conditions of validity together with (i) the display equation following (C.5), and (ii) Equation (M.21).

6.7 Image-forming: Summary of Formulae

For convenience, we summarise here the formulae obtained, within the simulation model, for the received voltage $E_n(t)$ and the image amplitude, for the various cases. First consider the uncorrelated case. For pointlike elements $E_n(t)$ is given by Equations (3.3) and (3.4). For solid elements, $E_n(t)$ is given exactly by (3.6). When the far-field-of-the-element approximation is invoked, that equation reduces to (3.7). For the subcase of a toneburst ($b = 0$), the expression reduces further to (3.11) (subject to certain conditions).

For both types of element, the in-phase image amplitude is then

$$\Pi(\mathbf{r}) = \sum_n w_n \sigma(\mathbf{r}, n) E_n \left[\left(r + |\mathbf{r} - \mathbf{R}_n| \right) / c \right] \tag{6.29}$$

(from Eqn 5.4). For each element type, E_n can be eliminated from (6.29) by substituting the appropriate formula from the previous paragraph, of course replacing t by $\left(r + |\mathbf{r} - \mathbf{R}_n| \right) / c$. Note that the abbreviation $\tau(\mathbf{r}, n) = \left(r + |\mathbf{r} - \mathbf{R}_n| \right) / c$ (Eqn 5.2) is often used in this report.

We turn to the correlated case. In the two equations for $E_n(t)$ in the first paragraph, one simply replaces, on the left side, E_n by E_{rn}, and on the right side, ξ by Y. In Equation (6.29), simply replace E_n by E_{rn}.

The equations so far described are all for the in-phase part. To form the corresponding equations for the quadrature part, add one subscript q on the right-hand side (i.e. to ξ, Y, E_n or E_{rn}, whichever is present) and one on the left (i.e. to E_n, E_{rn} or Π). The quadrature component for each of ξ, Y, E_n and E_{rn} is given in terms of the in-phase component by the Hilbert transform Equation (5.16). (However the same is not true of Π, since it has no time-dependence.)

The corresponding analytic signals are related to the component parts by $E_{an} = E_n + jE_{qn}$, together with similar equations for E_{ran}, ξ_a and Y_a (and, by special definition, Π_a). The total image amplitude Π_t is given by (5.21).

7. Quantisation and Related Effects

7.1 Quantisation and the Program's Procedure

Quantisation error or digitisation error is the rounding-off error introduced when a continuous quantity, in this case a sampled voltage, is represented in digital form—the

digital string, say of m bits, being of finite length. This error has been discussed, for example, by Proakis and Manolakis [1988] and briefly by Bateman and Yates [1988]. The special case of one-bit quantisation ('hard limiting'), in which only the value of the sign of the voltage is saved, has been discussed by Steinberg [1976] in the context of imaging. One- and two-bit quantisation have been used extensively in radio astronomy [e.g. Ables *et al.* 1975].

A related question is the effect of sampling, i.e. the fact that at each element the voltage is recorded at discrete[37] times only. The program does not address this effect, for it effectively assumes continuous sampling. However note that the Nyquist theorem tells us that, if the continuous signal is strictly confined to a certain bandwidth, that signal can be reconstructed from the samples provided that the sampling rate exceeds a certain critical value.

7.1.1 The Program's Quantisation Procedure

In this section we describe what the program does when the quantisation option is selected. Later (Section 7.2) it will be seen that, in the *uncorrelated* case, the effect of the program's procedure is very close to that in the operational system, but that in the *correlated* case there is a significant discrepancy.

The program's quantisation operation is performed, (i) in the uncorrelated case, on the voltage received at the n th element, $E_n(t)$, and its quadrature part $E_{qn}(t)$, and is performed, (ii) in the correlated case, on $E_{rn}(t)$ and $E_{rqn}(t)$. In all cases, the outputs (in-phase and quadrature) of the operation are passed on to the image-forming software described in Section 5.

The program's quantisation operation itself is described in Section N.1 of Appendix N. The basic feature is that the allowed outputs from the quantisation, 2^m in number, are uniformly spread and extend over some interval $(-E_{max}, E_{max})$ centred on zero voltage (Figure 15).

---

[37] In this report, 'discrete' times refers to a set of isolated times. We avoid the unfortunate meaning that has recently become common in engineering, whereby 'discrete' implies a single time.

$$\text{Case } m = 3 \quad (2^m = 8)$$
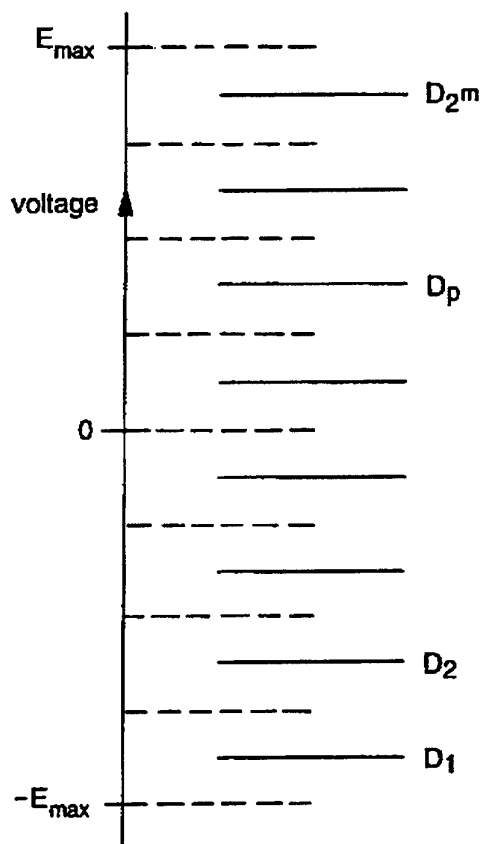
Figure 15: Quantisation. The uniformly quantised region $\left(-E_{max}, E_{max}\right)$ is subdivided into 2^m equal intervals. The allowed output values D_p are the midpoints of these intervals.

7.1.2 The Optimal Value of E_{max}

We define the *optimal* value of E_{max} as the value that leads to the best image quality for the scene (total target) concerned. The factors that determine this optimal value are discussed in Section N.2. That section also gives guidelines for selecting a good initial guess for the optimal E_{max}, to be refined thereafter by trial and error.

7.2 Limitations of the Program in Respect of Quantisation and Sampling

7.2.1 The Operational System

We first discuss the 'operational system', that is, the system—or class of systems—that the program is attempting to simulate. Here we are concerned only with part of the system, namely the sensors and the subsequent processing.

The causal flow of signals in this part of the operational system is as follows (Figure 16). (Some components, for example amplifiers and filters, have been ignored.)
1.   At each element, the pressure field produces an analogue voltage stream.
2.   The voltage stream is sampled at discrete times, with quantisation to m bits.
3.   a.   If the transmitted signal was a chirp or long coded pulse rather than a short pulse, the voltage stream is crosscorrelated with the projected signal (discrete correlation).[38]

b.   The quadrature part of the output of step 2 (uncorrelated case) or of step 3a (correlated case) is calculated (discrete Hilbert transform).
4.   Image-forming (including any weighting). As in Section 7.1.1, when the image-forming algorithm calls for the (dechirped) voltage at some given time, the system may interpolate between the sampling times or simply use the voltage value at the nearest sampling time.
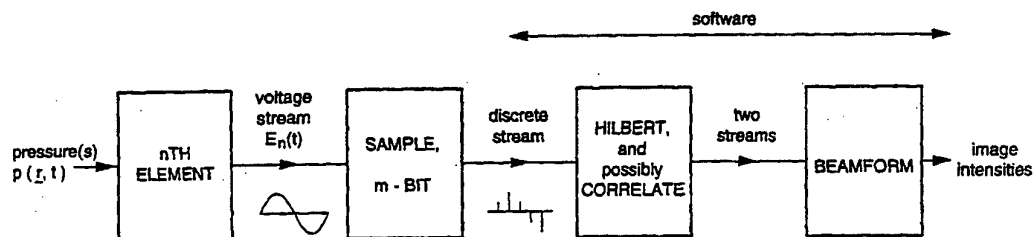
Figure 16:   Causal flow of signals in the receiver and signal processor of the operational system.

7.2.2 Limitations of the Program

The simulation program POINTSPR departs from the operational system described above in two ways.

---

[38] As discussed in Section 6.2.3.2, first, the crosscorrelation may be achieved by alternative, more efficient methods than numerical crosscorrelation and second, the steps a and b of 3. can be combined.

First, when there is *quantisation*, the program performs it, not before step 3 but after it. The result of this swapping depends on whether the signal is correlated. In the *uncorrelated* case, the only swapping is with the Hilbert transform operation, so that the in-phase image remains intact. Intuitively it appears that all 'reasonable' ways of obtaining an envelope from in-phase data will give very similar results; it is therefore expected that the simulation will be remain good in the uncorrelated case.

In the *correlated case*, the quantisation operation is being swapped also with the crosscorrelation operation. As a result the simulation of quantisation is likely to be mediocre, though the results should bear some resemblance to those from the operational system.[39]

Second, the program assumes continuous sampling. Hence it does not reproduce the effects of *sampling at discrete times* (step 2), with its consequence of either interpolating or 'rounding' to the nearest sampling time (step 4). To discuss the effects, consider first the case where the system does *not quantise* (m very large). Then the program accurately simulates discrete sampling, provided that two conditions are satisfied. The first condition is that the system's sampling rate exceeds the Nyquist frequency. The second is that the operational system's interpolation algorithm, used when image-forming, is a fairly good one. Next, consider the case where the system performs discrete sampling but *also quantises*. Then it *may* be the case that, under the two conditions just stated, the program's error is limited to the error that it has in treating quantisation.

7.3 A More Complete Simulation; Efficiency of Two Methods

The discussion of Section 7.2—on quantisation and sampling—prompts the question: How would one organise a program to more closely simulate an operational system (particularly from the sensors onwards)? This question will lead on to a discussion of the efficiency of the present method of calculation compared to a more close simulation.

To more fully simulate an operational system, one would need a program that is organised differently from POINTSPR. Its steps would follow very closely the steps of the operational system, set out in Section 7.2.1.

Thus, for a given set of target points, the sampled, quantised voltages in step 2 would be calculated *and stored* as voltage streams. Next, the crosscorrelation and Hilbert transform operations in step 3 would be performed as *discrete numerical* operations on the voltage stream (preferably in the frequency domain). Finally image-forming would be done, with either interpolation or the rounding of time to the nearest

---

[39] Besides the lack of a positive argument for 'goodness', there is the following negative argument. For the usual case of a long chirp, the system may be more tolerant of quantisation of the raw signal reflected from a given target (many samples) than of the correlated output pertaining to the same target (few samples), because averaging tends to eliminate errors; then the order of operations would make a big difference.

sampling point. One could include the simulation of noise (background acoustic and instrumental noise) by adding a term at step 2.

An interactive signal and image processing package along the above lines, TWODIMSY, has been developed for medical imaging at the Ultrasonics Laboratory, CSIRO (now part of the Division of Telecommunications and Industrial Physics, CSIRO) [Knight and Robinson 1987]. Features have been added since that time to accommodate AMI. In that package, processing steps can be made to occur in the desired order by high-level programming.

It is of interest to compare the computing times, and hence the efficiencies of: (i) the method of the present report, and (ii) the stored voltage method. It might be thought that, in the case of a chirp, method (i) is superior, because no crosscorrelation needs to be performed. The latter feature arises because the corresponding autocorrelation has already been performed analytically for each point target. The truth turns out to be more complicated.

The number of floating-point operations needed to produce an image is roughly:

$$10(N_t + 1)N_v N_e \qquad \text{method of present report}$$
$$[10N_t + 10N_s \log_2 N_s + 10N_v]N_e \qquad \text{stored voltage method}$$

(7.1)

Here N_t is the number of point targets, N_v is the number of voxels to be calculated, N_e is the number of elements and N_s is the number of time samples in the vector of voltages to be dechirped; the numerical coefficients in (7.1) are rubbery. Inside the square bracket *the second of the three terms is to be included only when* crosscorrelation is to be performed.

N_s is given by

$$N_s = f_s S,$$

where f_s is the sampling frequency and S is the length in time of the stream to be dechirped. cS must be at least as great as the length of the chirp or toneburst plus *twice* the span of ranges being observed.

In the *correlated* case, let us simplify the expression for the stored voltage method as follows. *Inside the logarithm,* we put $N_s = 2^{16} = 65\,536$; the logarithm will then be represented semiquantitatively for N_s lying anywhere in the interval 2^{11} to 2^{24} (say). Then the second term in the square bracket reduces to

$$160N_s = 160 f_s S.$$

Let N_r be the number of range voxels being observed. Recall that a range voxel is $c/2B$ (from Eqn 2.7 with 0.886 rounded to unity.) From above, taking S to be as small as possible for the given span of ranges, and assuming the chirp length to be small compared to cS, we have $N_r = (cS/2)/(c/2B) = BS$. The second term in the square brackets becomes

$$160 f_s (N_r / B).$$

Thus, for example, if $f_s / B = 27$ — not atypical of AMI designs — the result for the stored voltage method finally reduces to

$$[10N_t + 4300N_r + 10N_v]N_e. \qquad (7.2)$$

For an *uncorrelated* signal, from (7.1) with the 'log' term dropped, it is readily seen that, provided $N_t \ll N_v$, the stored voltage method has a speed advantage given by the ratio $N_t + 1:1$. Note however that in the case of a *single target* this ratio is only 2; then the performance of the present method falls not far short.

For a *correlated* signal, *there is a critical value of* N_t *below which the present method is faster* and above which the stored voltage method is faster. Thus, at least in the correlated case, the two methods are complementary. The critical value is obtained by equating the two expressions in (7.1), invoking (7.2) and again assuming $N_t \ll N_v$. Consider as an example the case where the received stream occupies 3000 range voxels (a 3 m range interval subdivided into 1 mm range voxels) and the number of voxels to be calculated is 120 000 (3 planar plots of 200 X 200). Then the critical value of the number of targets is 11.

By way of discussion, it has been noted that the present method (method i) has an advantage regarding crosscorrelation calculations. Despite this, the method 'loses' when there are many targets. The reason for this is seen from the first line of (7.1). The method requires image-forming to be done, not just for each voxel, but for each combination of a *point target* and a voxel.

8. Sample Graphs

As examples of the use of the program, we present a number of graphs obtained using POINTSPR and the plotting routines. We first present linear graphs (intensity along a line or curve) and then proceed to planar or surface graphs. Viewgrids for the image were discussed in Section 4.

8.1 Linear Graphs

Four data sets, named r1 to r4, were used to obtain the sample linear graphs. These data sets are given in Table 1 (containing the parameters common to the four data sets) and Table 2 (the differing parameters). A common feature of the data sets is the receiving array: generated as a random array, it is one specific realisation used repeatedly. This random array is very sparse: at each site of this $\lambda/2$ array, the probability of the site's being occupied is 0.15%. The expected number of elements is 6000; the actual number is 5917. Also common is the single target, located in the broadside direction at a range of 1.0 m.

Table 1: Parameters common to the linear graphs.

Parameter	Value
(a) Array and Target Parameters	
central frequency f_c	3 MHz
grid spacing d	$\lambda_c/2 = 0.25$ mm
grid points along a side M	2000
occupation pattern	random
specific array	constant (original seed vector)
probability of occupation	0.15%
element type	pointlike
spherical spreading	correct for
quantisation	none
shading	unshaded
number of targets	1
target:	
range	1.0 m
elevation	0.0
azimuth	0.0
target strength	1.0
(b) Viewgrid Parameters	
plot type	linear
viewtype	radially curved
(θ, ϕ) rotated?	no
secondary origin	at target point
viewwindow size:	
δ	0.220 m
ε	0.220 m
ζ	0.055 m
normalisation for plotting	to value at first target
procedural responses:	
Monte Carlo?	no
number of viewgrid points?	yes, change from default
viewwindow widths?	yes, change from default
low-resolution option?	no
special facilities in POINTSPR?	no
special facilities in PLINE?	no

Table 2: Parameters varying between the linear graphs.

	unit	r1		r2		r3		r4	
(a) Array and Target Parameters									
pulse description		long toneburst		medium toneburst		short toneburst		correlated chirp	
bandwidth B	MHz	0		0		0		1.0	
pulse duration T in cycles of f_c	cycles	600		100		3		100	
(b) Viewgrid Parameters									
axis to be plotted		δ	ζ	δ	ζ	δ	ζ	δ	ζ
centre of viewwindow:									
range	m	1.0	at	as					
elevation	rad	0.0	tar-						
azimuth	rad	0.090	get						
number of viewgrid points:									
δ		4000	3						
ε		3	3						
ζ		3	4000						
calcul. sidelobe average?		yes	no			on			
centre of exclusion region (def't=secondary origin)		de-fault	n.a.						
fraction of p'ts to exclude		25%	n.a.						
procedural responses:									
move centre of viewwindow?		yes	no						
move origin of coordinates on graph?		no	n.a.					left	

The four data sets differ in respect of the pulse: in the sets r1 to r4 respectively the pulse is a long toneburst, a medium-length toneburst, a short toneburst and a correlated chirp (Table 2). Their characteristics are chosen to illustrate different behaviours as follows. The long toneburst is chosen long enough so as to behave as a continuous wave (cw) in the context of the given array. The duration of the short toneburst and the bandwidth of the correlated chirp are both chosen to give a range resolution of about 1 mm. The medium toneburst is chosen so as to contain quite a large number of cycles, but nevertheless to depart noticeably from a cw in its azimuthal beam pattern.

The resulting plots along the δ (azimuth) axis and the ζ (range) axis are shown in Figures 17 and 18 respectively. In each of the two figures, the graphs from the sets r1 to r4 are shown simultaneously. In Figure 17, an off-centre view was chosen in order to display more detail.
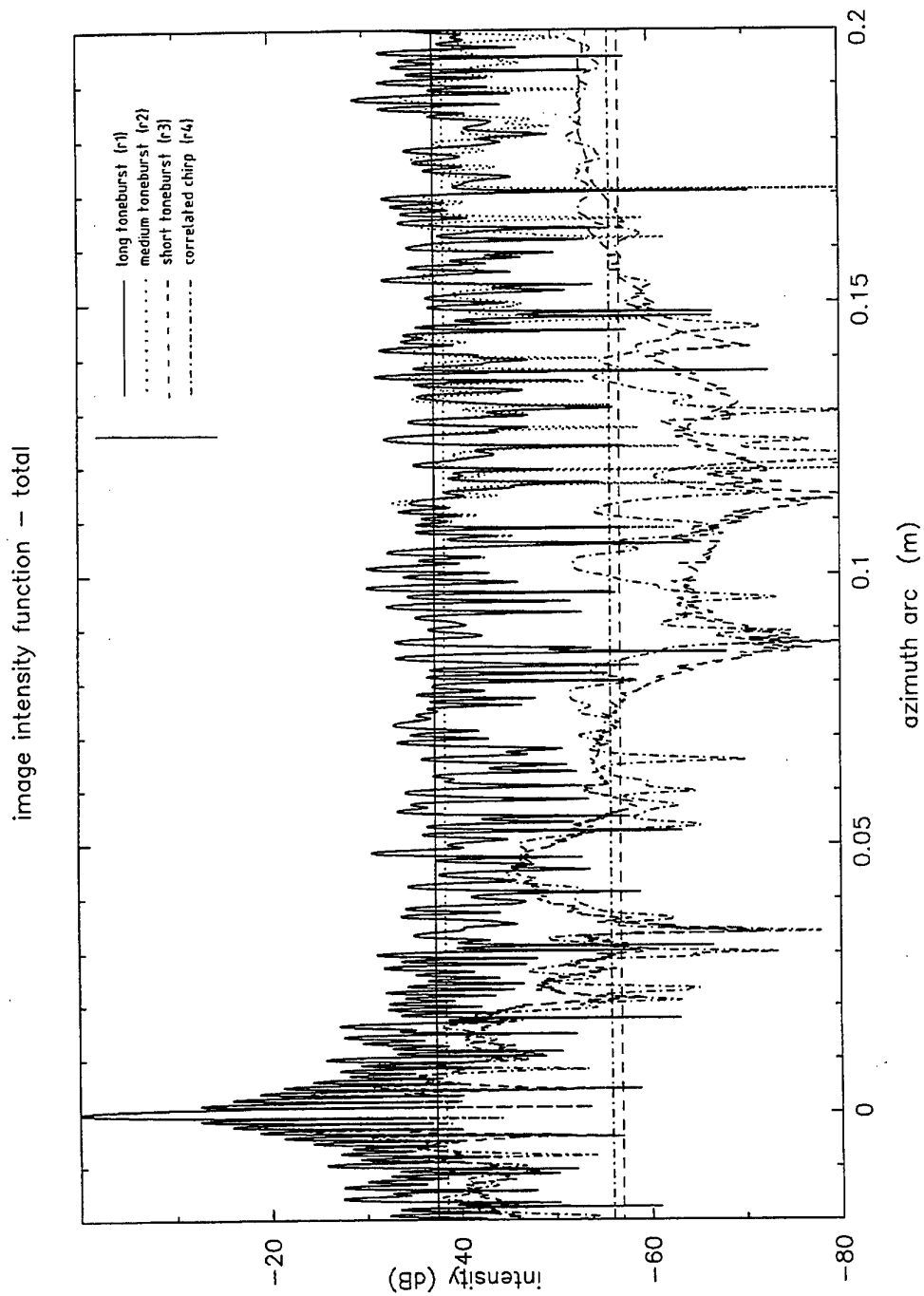
image intensity function — total

Figure 17: Linear graphs giving image intensity (decibel scale) along the arc of increasing azimuth through the target point, for the data sets r1 to r4 given in Tables 1 and 2. The four horizontal lines are the respective sidelobe averages (see Section 4.4.1).
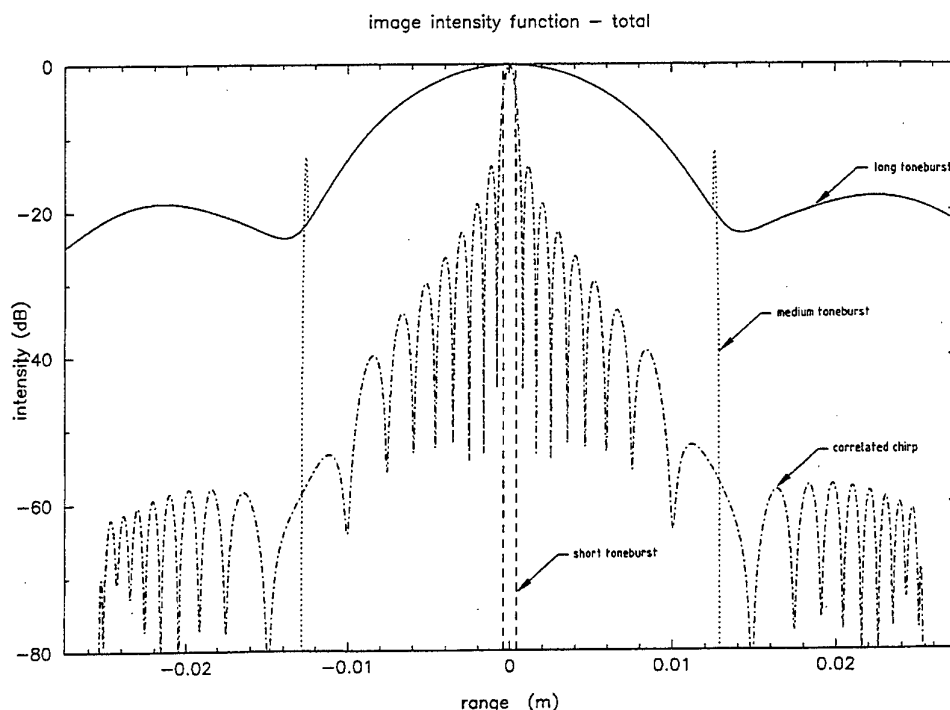
image intensity function — total

Figure 18: As for Figure 17, but along a radial line through the target point. The range plotted is the range offset from the target. (No sidelobe averages are plotted.)

Some interesting features emerge. In the azimuth plot (Fig. 17), the curve for the medium toneburst begins to depart from that for the long toneburst at a definite azimuth arc coordinate of 0.103 m. The short toneburst and the correlated chirp (both with range resolution 1 mm) behave quite similarly as a function of azimuth, especially in respect of their sidelobe averages. In the plot against range (Fig. 18), the four graphs are all quite different. For the short toneburst, outside a narrow interval the intensity is zero; the rectangular shape reflects the rectangular shape of the envelope of the pulse. The correlated chirp has a shape reflecting that of the pulse's autocorrelation function Y (Eqn 6.19) (rather, the square of Y's envelope). The intensity for the long toneburst falls off slowly; the detail accords reasonably well with the standard theory of the depth of field [Steinberg 1976, p. 52]. For the medium burst, the amplitude coincides with that of the long burst out to a critical range on each side; then, after a rapid rise and fall, it becomes zero.

8.2 Planar Graphs

Four data sets, named r15 to r18, were used to obtain the sample planar graphs. These sets are given in Table 3 (parameters common to the four data sets) and Table 4 (the differing parameters). It will be seen that there is a very high degree of commonality

between the sets, as all the sets are selecting parts of the same point spread function. The array used is square and regular. The waveform is a correlated chirp. The single target is placed at an azimuth of +45° and a range of 1.0 m.

The four plots (Table 4) differ in respect of the orientation of the surface over which the amplitude is to be plotted, the scale (whether a high- or low-resolution graph is displayed), and whether the image amplitude is represented by false colour or by a 3-D mesh plot. Here the 'low-resolution' graph refers to an option that selects automatically an interval of azimuth and/or elevation that embraces a large number of sidelobes along each angular axis.

Figure 19 plots the results from data set r15; it gives, in false colour, a high-resolution plot over the $\delta\varepsilon$ surface (spherical surface with range = constant). The dominant features are the main lobe and sidelobes; these lobes, at least along the two axes, resemble the lobes of the corresponding far-field, monofrequency system, especially in respect of position. The separations of the sidelobes in the two principal directions are in the ratio $\cos 45°$, as expected.

Table 3: Parameters common to the planar graphs.

Parameter	Value
(a) Array and Target Parameters	
central frequency f_c	5 MHz
pulse description	correlated chirp
bandwidth B	0.8333 MHz
pulse duration T	60 cycles of f_c
grid spacing d	$10\lambda_c = 3.00$ mm
grid points along a side M	100
occupation pattern	filled (hence square array)
element type	pointlike
spherical spreading	correct for
quantisation	none
shading	unshaded
number of targets	1
target:	
range	1.0 m
elevation	0.0
azimuth	$\pi/4$ rad
target strength	1.0
(b) Viewgrid Parameters	
plot type	planar
viewtype	radially curved
centre of viewwindow	at target (default)
normalisation for plotting	to value at first target
calculate sidelobe average?	no
procedural responses:	
Monte Carlo?	no
number of viewgrid points	yes, change from default
viewwindow widths	yes, change from default
low-resolution option?	no
special facilities in POINTSPR?	no
special facilities in PLPLANE?	no (or n.a. in PLMESH cases)

Table 4: Parameters varying between the planar graphs.

	unit	r15	r16	r17	r18
(a) Array and Target Parameters					
nil					
(b) Viewgrid Parameters					
surface to be plotted		$\delta\varepsilon$	$\delta\varepsilon$	$\delta\zeta$	$\delta\zeta$
representation		false colour	mesh	mesh	false colour
viewwindow size:					
δ	m	0.010	as		0.10
ε	m	0.010		for	0.10
ζ	m	0.020		r15	0.024
number of viewgrid points (before cropping):					
δ		301	111	161	301
ε		301	111	3	3
ζ		3	3	161	301
in case of mesh:					
decibels?			no	no	
crop?			no	to 60% in both directions	
transpose axes?			no	yes	
procedural responses: special facilities in PLMESH?			no	yes	

Figure 20 plots the results from data set r16. The figure shows the same output data[40] as Figure 19, but plotted as a mesh in 3-D. The plot is of the image amplitude on a linear scale. A positive feature of the mesh plot, compared to false colour, is that the whole continuous range of possible image amplitudes is available for display.

---

[40] Actually fewer viewgrid points have been used for the mesh plot, to improve its clarity.
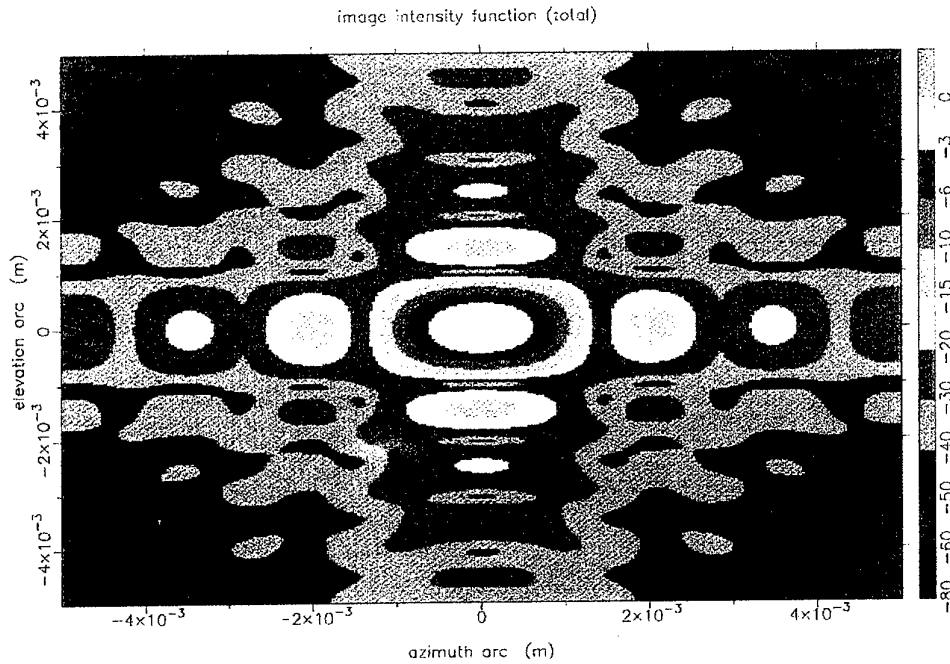
Figure 19: *Planar plot in false colour, showing the image intensity over a spherical surface through the target, for the data set r15 (Tables 3 and 4). The intensity units are decibels. Azimuth arc length is measured from the target.*
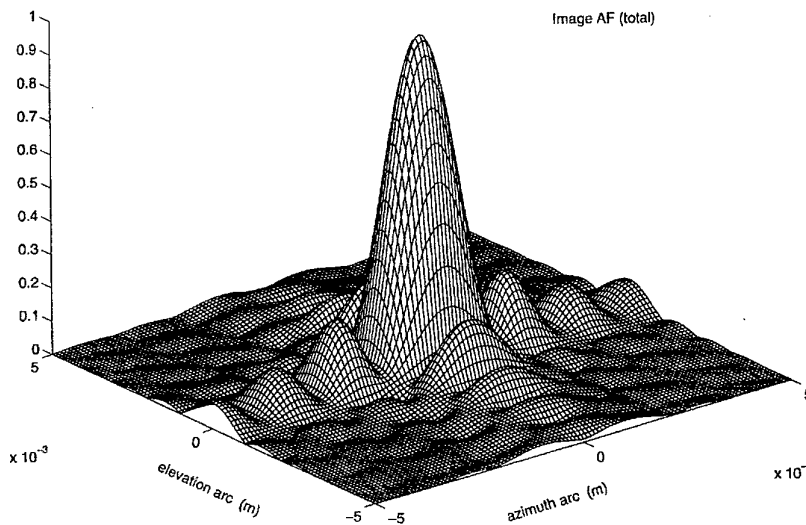
Figure 20: *Planar plot, represented as a 3-D mesh, showing the image intensity over a spherical surface through the target, for the data set r16 (Tables 3 and 4). The quantity plotted is the amplitude (not intensity or number of decibels).*

Figure 21: *Planar plot, represented as a 3-D mesh, showing the image intensity over a conical surface of constant elevation through the target, for the data set r17 (Tables 3 and 4).*
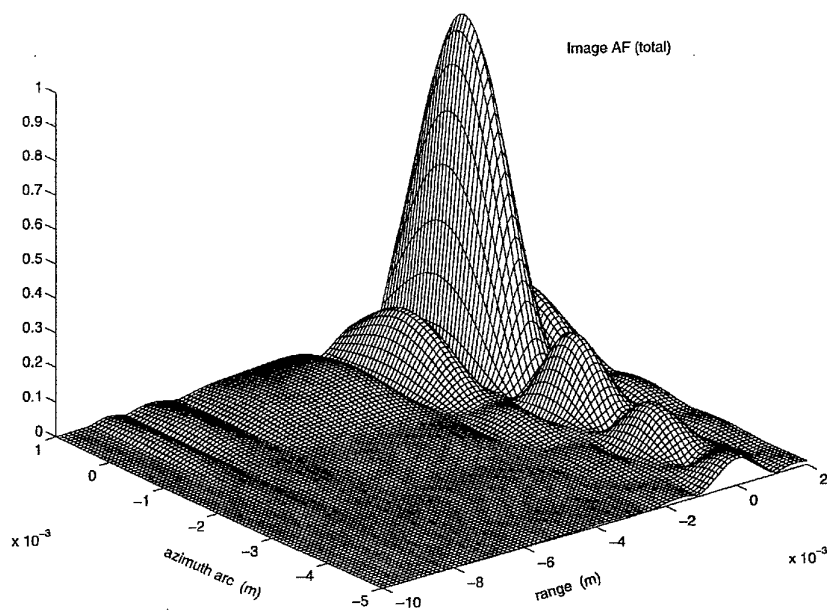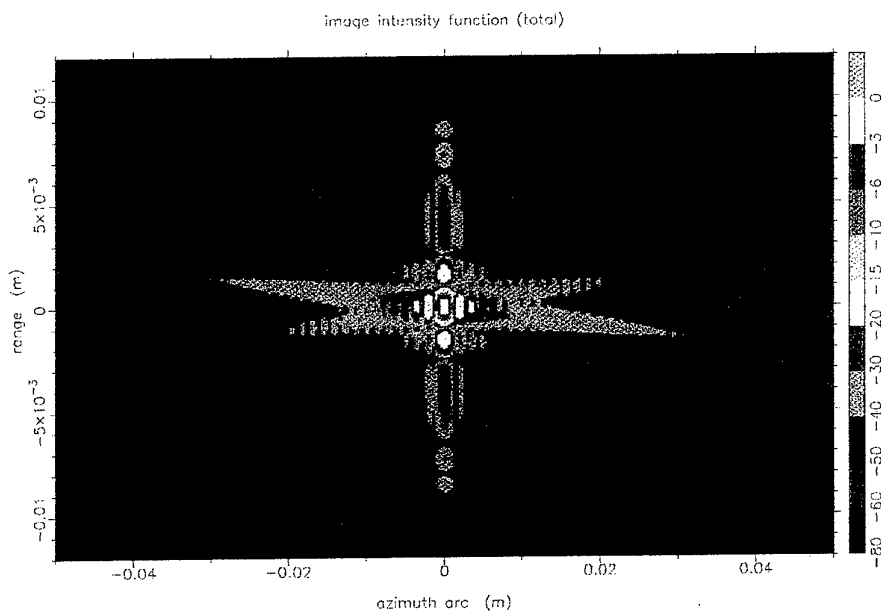

Figure 22: *Planar plot in false colour, showing the image intensity over a conical surface of constant elevation through the target, for the data set r18 (Tables 3 and 4).*

Figure 21 (data set r17) is a similar plot to Figure 20, except that it shows the $\delta\zeta$ surface. The values of range extend out to a little beyond the value (9.0 mm from the target) at which the predicted image amplitude becomes (and stays) precisely zero (Section 5.1.3).[41] The difference in shape between the azimuth sidelobes and the range sidelobes is clearly seen.

Our experience is that the mesh plots can produce an excellent display of the results, at least in some cases. A possible drawback is that mesh plots require extra work; an example of this is given in the previous footnote. A second possible drawback with mesh plots is that important items in the plot are sometimes hidden. Incidentally, MATLAB's algorithm for hidden line removal sometimes fails. Third, the mesh plots may not work so well with graphs containing many sidelobes (low-resolution graphs). This is because, first, a wide dynamic range (perhaps 80 dB) needs to be represented, and second, the lobes are very narrow.

Figure 22 (data set r18) extends the graph of Figure 21 to greater angular displacements from the target (low-resolution graph), and represents the image in false colour. The inner lobes of the two plots— including all the range sidelobes— can be compared figure-to-figure. The graph clearly shows the umbra-penumbra distinction discussed in Appendix E. (Black represents an intensity less than -80 dB—often a strictly zero intensity.) The V-shaped boundaries at the top and bottom conform closely to Equation (E.3).

9. Conclusion

This report has described a program, POINTSPR, that simulates underwater acoustic imaging. Sample inputs and outputs from the program have been presented. The report gives analytic results from the simulation model—for example, the shape of the range sidelobes and the response of extended elements to a chirp. The detailed theory of three aspects of real acoustic imaging systems has been given: dechirping, demodulation, and image-forming. The program gives useful simulations over a considerable range of conditions. The limitations of the program, however, have been noted; and a brief description has been given of the architecture of a program that would enable more general simulations.

10. Acknowledgements

---

[41] In order to show greater resolution: (i) the graphs have been cropped, (ii) the graphs have ceased to be centred at the peak, and (iii) the δ and ζ axes have been interchanged.

Thomson Marconi Sonar and CSIRO Division of Telecommunications Physics, exemplified by Ian G. Jones and David Robinson, contributed greatly to the success of the mine imaging program during the period of this work.

11. References

Bahl, R. and Powers, J.P. (1990). *Computer Model of a High-Resolution Imaging Sonar* (Technical Report No. NPS62-90-011). Monterey, Calif.: Naval Postgraduate School.

Bateman, A. and Yates, W. (1988). *Digital Signal Processing Design.* London: Pitman.

Bergland, G.D. (1969). A Guided Tour of the Fast Fourier Transform. *IEEE Spectrum*, 6, pp. 41-52.

Blair, D.G. (1993). *Array Design: Literature Survey for a High-Resolution Imaging Sonar System — Part 1* (MRL Technical Note MRL-TN-658). Melbourne: Materials Research Laboratory.

Blair, D.G. (1997). *Underwater Acoustic Imaging: A Computing Hardware Approach to Rapid Processing* (DSTO Technical Note DSTO-TN-0099). Melbourne: Aeronautical and Maritime Research Laboratory.

Blair, D.G., Bedwell, I., Anstee, S.D. and Li, Y. (1994). Use of a Random Array for High-Resolution Underwater Acoustic Imaging. *International Conference on Underwater Acoustics*, University of New South Wales, 5-7 Dec. 1994, pp. 56-57. Darlinghurst, N.S.W.: Australian Acoustical Society.

Blair, D.G. and Jones, I.S.F. (1998). *Underwater Acoustic Imaging: Rapid Signal Processing* (DSTO Technical Note DSTO-TN-0098). Melbourne: Aeronautical and Maritime Research Laboratory.

Blair, D.G., Thompson, J. and S. Anstee (1997). The Effectiveness of Sparse Random Arrays for Underwater Acoustic Imaging. *Fifth International Congress on Sound and Vibration*, University of Adelaide, 15-18 Dec. 1997, p. 65. Adelaide: Australian Acoustical Society.

Bottoms, A., Eagle, J. and Bayless, H. (Eds) (1995). *Proceedings of the Autonomous Vehicles in Mine Countermeasures Symposium* (4-7 April 1995). Monterey, Calif.: Naval Postgraduate School.

Burdic, W.S. (1991). *Underwater Acoustic System Analysis*, 2nd Ed. Englewood Cliffs, New Jersey: Prentice-Hall.

Clay, C.S. and Medwin, H. (1977). *Acoustical Oceanography: Principles and Applications.* New York: John Wiley.

Cuschieri, J.M. and Cao, P. (1992). 3-D Sonar System for UUVs. *Sea Technology,* **33**, Number 12, pp. 47-53.

Ditchburn, R.W. (1952). *Light.* Vol. I of: *The Student's Physics.* London: Blackie.

Fry, F.J. (Ed.) (1978). *Ultrasound: Its Applications in Medicine and Biology—Parts 1 and 2.* Amsterdam: Elsevier Scientific.

Hill, C.R. (Ed.) (1986). *Physical Principles of Medical Ultrasonics.* Chichester, England: Ellis Horwood.

Jenkins, F.A. and White, H.E. (1950). *Fundamentals of Optics.* New York: McGraw-Hill.

Jones, I.S.F. (1994). High Resolution Underwater Imaging. *International Conference on Underwater Acoustics,* University of New South Wales, 5-7 Dec. 1994, p. 40. Darlinghurst, NSW: Australian Acoustical Society.

Jones, I.S.F. (1996). *Underwater Acoustic Imaging Innovation Program* (DSTO Technical Note DSTO-TN-0065). Melbourne: Aeronautical and Maritime Research Laboratory.

Kino, G.S. (1987). *Acoustic Waves: Devices, Imaging, and Analog Signal Processing.* Englewood Cliffs, N.J.: Prentice-Hall.

Knight, P.C. and Robinson, D.E. (1987). An Approach to Digital Signal Processing for Ultrasonics Research. *Ultrasound in Med. and Biol.,* **13**, No. 6, 345-352.

Knudsen, D.C. (1989). A New Beamformer for Acoustic Imaging. In: *Oceans '89: An International Conference Addressing Methods for Understanding the Global Ocean.* New York: IEEE Press.

Knuth, D.E. (1981). *Seminumerical Algorithms,* 2nd Ed. Vol. 2 of: *The Art of Computer Programming.* Reading, Mass.: Addison-Wesley.

Neubauer, W.G. (1986). *Acoustic Reflection from Surfaces and Shapes.* Washington, D.C.: Naval Research Laboratory.

Ogilvy, J.A. (1991). *Theory of Wave Scattering from Random Rough Surfaces.* Bristol, England: Adam Hilger.

Perley, R.A., Schwab, F.R. and Bridle, A.H. (Eds.) (1989). *Synthesis Imaging in Radio Astronomy: a Collection of Lectures from the Third NRAO Synthesis Imaging Summer School.* San Francisco: Astronomical Society of the Pacific.

Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1986). *Numerical Recipes: The Art of Scientific Computing.* Cambridge, England: Cambridge Univ. Press.

Proakis, J.G. and Manolakis, D.G. (1988). *Introduction to Digital Signal Processing.* New York: Macmillan.

Proakis, J.G. and Manolakis, D.G. (1992). *Digital Signal Processing: Principles, Algorithms, and Applications,* 2nd Ed. New York: Macmillan.

Rihaczek, A.W. (1985). *Principles of High-Resolution Radar,* Revised Version. Los Altos, Calif.: Peninsula.

Robinson, D.E. and Wing, M. (1984). Lateral Deconvolution of Ultrasonic Beams. *Ultrasonic Imaging,* **6,** pp. 1–12.

Smith, S.W., Pavy, H.G., and von Ramm, O.T. (1991). High-Speed Ultrasound Volumetric Imaging System—Part I: Transducer Design and Beam Steering. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control,* **38,** pp. 100-108.

Steinberg. B.D. (1976). *Principles of Aperture and Array System Design—Including Random and Adaptive Arrays.* New York: John Wiley.

Steinberg, B.D. and Subbaram, H.M. (1991). *Microwave Imaging Techniques.* New York: John Wiley.

Thuraisingham, R.A. (1994a). *Models to Estimate High Frequency Acoustic Scattering due to Thermal Fine- and Micro-structure of the Ocean* (DSTO Research Report DSTO-RR-0001). Melbourne: Aeronautical and Maritime Research Laboratory.

Thuraisingham, R.A. (1994b). *Theoretical Estimates of High Frequency Acoustic Attenuation and Backscattering from Suspended Sand Particles in the Ocean and in an Estuary* (DSTO Technical Report DSTO-TR-0078). Melbourne: Aeronautical and Maritime Research Laboratory.

Truver, S.C. (1995). Naval Mine Countermeasures: Lagging Behind the Threat? *International Defense Review,* **28,** No. 9, pp. 54–59.

Turner, R.E. and Betts, D.S. (1974). *Introductory Statistical Mechanics.* London: Chatto and Windus, for Sussex Univ. Press.

Urick, R.J. (1983). *Principles of Underwater Sound,* 3rd Ed. New York: McGraw-Hill.

Verveur, J. (1990). Mine Warfare: Threats and Countermeasures. *Defence Systems International: Sea Systems 1990,* pp. 173–176.

Webb, S. (Ed.) (1988). *The Physics of Medical Imaging*. Bristol, England: Adam Hilger.

Ziomek, L.J. (1985). *Underwater Acoustics: a Linear Systems Theory Approach*. New York: Academic Press.

Appendix A: The Two-Nodes Constraint for a Correlated Waveform

Consider a chirp pulse of many cycles. Then the two-nodes recommendation would require that, for a given f_c, we choose b and T so that there is a pair of integers, m_1 and m_2, such that both

$$-2\pi f_c(T/2) + b(T/2)^2 = m_1\pi + \pi/2$$
$$2\pi f_c(T/2) + b(T/2)^2 = m_2\pi + \pi/2.$$

(A.1)

Alternatively, we may choose to simulate a waveform that is (2.1) but with a sine function; in that case both $\pi/2$ terms in (A.1) would be dropped. Actually the recommendation or requirement (A.1) can be ignored, for two reasons. First, numerical results in Section 6.5.2 will show that, in the correlated case, the step function does not lead to any unfortunate *qualitative* effects, and the unfortunate *quantitative* effects are hardly noticeable. The second reason is that, even if the two-nodes requirement were to lead to unfortunate effects, because of the pulse contains many cycles, the pair of parameters b and T input into the program are very close to another pair (b,T) that *does* satisfy the requirement. Thus, apart from minor shifts in the output graphs (of quantities plotted against time), the *latter* pair is modelled well by those graphs.

Appendix B: Response of Solid Elements: Uncorrelated Case

B.1. General

From Equation (3.6), the voltage produced by the solid element is, without approximation,

$$E_n(t) = K_2 K_s \sum_j \frac{a_j}{r_j} \iint_{S_n'} \frac{1}{|\mathbf{r}_j - \mathbf{R}_n - \mathbf{R}'|} \xi \left[t - \left(r_j + |\mathbf{r}_j - \mathbf{R}_n - \mathbf{R}'| \right) / c \right] dX' \, dY' . \tag{B.1}$$

Here $\mathbf{R}' = (X', Y')$ is the location of a transducing point on S_n', with the origin of $\mathbf{R}'$ chosen at the centre $(\mathbf{R}_n)$ of the element (see Fig. 23). Invoking the far-field-of-the-element assumption, in the denominator we replace $\mathbf{R}'$ by zero, which enables the inverse distance factor to be taken outside the integral. Within the argument of ξ we may use the far-field expression (see Fig. 23)

$$\begin{aligned}
|\mathbf{r}_j - \mathbf{R}_n - \mathbf{R}'| &= |\mathbf{r}_j - \mathbf{R}_n| - (\mathbf{r}_j - \mathbf{R}_n) \cdot \mathbf{R}' / |\mathbf{r}_j - \mathbf{R}_n| \\
&= |\mathbf{r}_j - \mathbf{R}_n| - (X' u_{jn} + Y' v_{jn}) ,
\end{aligned} \tag{B.2}$$

with u_{jn} and v_{jn} given by (3.9). Thus the result (3.7) is obtained.
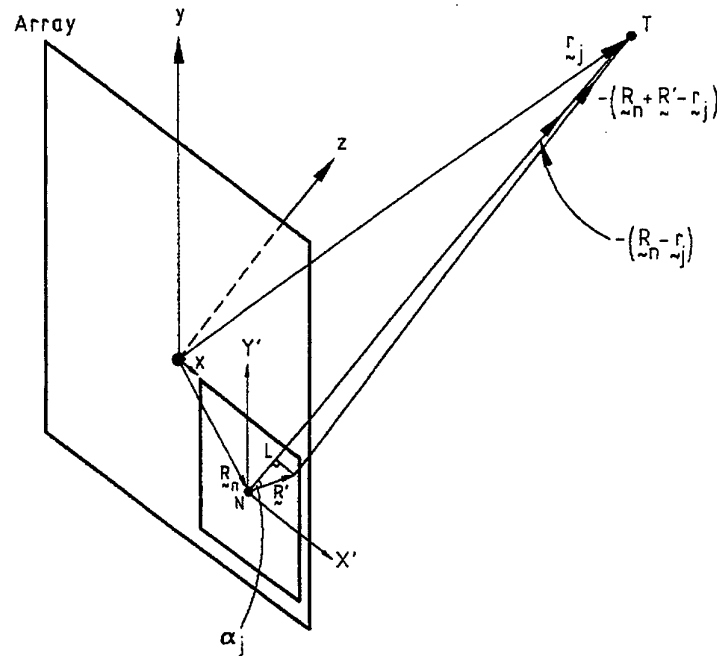
Figure 23: Showing displacements and angles relevant to a solid element. The n th solid element, shown as a square, has its centre N at a displacement $\mathbf{R}_n$. From there, a typical transducing point is displaced by $\mathbf{R}'$. The software of the operating system assumes that the target point T is in the far field of the element. The last term in the last line of Equation (B.2) is the length NL. For tilelike elements, a similar figure applies but with $\mathbf{R}'$ replaced by $\mathbf{R}_{nk}$.

B.2. Case of Toneburst

In Equation (B.1), $\xi(\cdot)$ is given by (2.1). When we substitute $b = 0$, put $\xi(\cdot)$ equal to the real part (Re) of its complex form and recall that $\xi(\cdot)$ is truncated at $\pm T/2$, Equation (B.1) becomes

$$E_n(t) = K_2 K_s \sum_j \frac{a_j}{r_j |\mathbf{r}_j - \mathbf{R}_n|} \text{Re} \exp\left\{ j2\pi f_c \left[t - \left(r_j + |\mathbf{r}_j - \mathbf{R}_n| \right)/c \right] \right\}$$

$$\times \iint_{S'_n} \exp\left[j2\pi f_c \left(X' u_{jn} + Y' v_{jn} \right)/c \right] \theta(t, \mathbf{R}') \, dX' \, dY',$$

(B.3)

where

$$\theta(t, \mathbf{R}') = 1 \qquad \left| t - \left[r_j + \left| \mathbf{r}_j - \mathbf{R}_n \right| - \left(X' u_{jn} + Y' v_{jn} \right) \right] \middle/ c \right| < T/2$$

$$= 0 \qquad \text{otherwise}.$$

(B.4)

At this stage we approximate by dropping the term in u and v in parentheses from the condition in (B.4). Then $\theta(t, \mathbf{R}')$ may be moved in front of the integral sign in (B.3). The integral then becomes the product of two integrals, each of which is straightforward. The final result obtained for the voltage is (3.11), where D_s is the directivity given by (3.10). In the step from (B.3) to the final result, essentially the integral (without θ) becomes the directivity, while the θ factor combines with the exponential in front of the integral to yield the ξ factor in the result, because of (2.1).

The approximation concerning θ should be satisfactory provided that the dropped term is always small compared to T. For then, although θ is sometimes given the wrong value, this is in a sense a rare event. The stated condition may be rewritten thus:

$$R' \left(\cos \alpha_j \right) \middle/ c \ll T,$$

(B.5)

where $\alpha_j = \left(\mathbf{r}_j - \mathbf{R}_n, \mathbf{R}' \right)$ (i.e. the angle between the two vectors) (Fig. 23). Inserting in (B.5) the maximum value of R', and the direction of $\mathbf{R}'$ that yields the minimum value of α_j (namely, when $\mathbf{R}'$ lies along the projection of $\mathbf{r}_j - \mathbf{R}_n$ onto the plane of the array), we obtain the sufficient condition

$$\frac{M'd}{\sqrt{2}} \sin\left(\mathbf{r}_j - \mathbf{R}_n, \hat{z} \right) \ll cT.$$

(B.6)

Using the fact that (at least approximately)

$$\sin\left(\mathbf{r}_j - \mathbf{R}_n, \hat{z} \right) \le 2 \max\left[\sin\left(\mathbf{r}_j, \hat{z} \right), \frac{M'd/\sqrt{2}}{r_j} \right],$$

the following sufficient conditions are obtained:

$$\max_j \left[\frac{M'd}{\sqrt{2}} \sin\left(\mathbf{r}_j, \hat{z} \right) \right] \ll cT$$

$$\max_j \left[\frac{(M'd)^2}{2r_j} \right] \ll cT.$$

(B.7)

(Since every target j must satisfy this condition, the 'max' is inserted in each of the conditions B.7.) Note that these conditions arose from the requirement that a wave from the jth target sweep through any given element in a time short compared to the duration of the toneburst. From (B.7), it is required that all the targets be sufficiently close to broadside (the first of the two conditions) and sufficiently far away. Note the qualification 'sufficiently': for example, what counts as a sufficiently small angle with broadside depends on other parameters such as T.

Appendix C: Response of Tilelike Elements: Uncorrelated Case

C.1. General

For tilelike elements, of linear size $M'd$, each element is a square array of subelements; these subelements are identical to the pointlike elements discussed in Section 3.2.1. A subelement is placed at each of the *original* grid points that lie within the tilelike element; thus their spacing is d. There are M'^2 subelements in the element. Another feature is that internally the tilelike element is *far-field steered*. This means that two assumptions are made. First, that the operational system does some preliminary image-forming (see Section 5.1.1) using the outputs from the subelements within the element. Second, that in doing so, for the image point of interest, the operational system introduces a time-delay for each subelement as calculated from the formulae appropriate to the far field *of the element*.

The output voltage of the element is therefore

$$E_n(t) = K_6 \sum_k p_{scat}(\mathbf{R}_n + \mathbf{R}_{nk}, \ t - t_{nk}),$$ (C.1)

where $\mathbf{R}_{nk}$ is the location of the kth subelement relative to the centre of the nth element (at $\mathbf{R}_n$). In (C.1), t_{nk} is the delay to be applied to the signal at the kth subelement, relative to the centre of the element. If the steering applied were *not* far-field, this delay would be

$$t_{nk} = \left(-|\mathbf{R}_n + \mathbf{R}_{nk} - \mathbf{r}| + |\mathbf{R}_n - \mathbf{r}|\right)\Big/c.$$

Since the steering *is* far-field, the delay is modified as follows. Let α be the angle

$$\alpha = (\mathbf{r} - \mathbf{R}_n, \mathbf{R}_{nk}),$$ (C.2)

where $\mathbf{r}$ is the location of the *image point*. Then from a diagram identical to Figure 23 but with some relabelling (the target point T, $\mathbf{r}_j$, α_j and $\mathbf{R}'$ are replaced respectively by the image point P, $\mathbf{r}$, α and $\mathbf{R}_{nk}$; also LN $= ct_{nk}$) it is seen that

$$t_{nk} = (R_{nk} \cos\alpha)\big/c = \frac{\mathbf{R}_{nk} \cdot (\mathbf{r} - \mathbf{R}_n)}{c|\mathbf{r} - \mathbf{R}_n|}.$$ (C.3)

The voltage output, in terms of the transmitted signal, is obtained by substituting the pressure from (2.3), along with Equation (C.3), into (C.1). The formula used by the program, however, makes the *second* far-field assumption, that the *physical* development of the reflected wave proceeds as though each target were in the far field of the element. Thus the voltage as calculated is

$$E_n(t) = K_2 K_6 \times$$

$$\sum_j \frac{a_j}{r_j |\mathbf{r}_j - \mathbf{R}_n|} \sum_k \xi \left\{ t - \left[r_j + |\mathbf{r}_j - \mathbf{R}_n| - \frac{(\mathbf{r}_j - \mathbf{R}_n) \cdot \mathbf{R}_{nk}}{|\mathbf{r}_j - \mathbf{R}_n|} + \frac{(\mathbf{r} - \mathbf{R}_n) \cdot \mathbf{R}_{nk}}{|\mathbf{r} - \mathbf{R}_n|} \right]\Big/c \right\}$$ (C.4)

For tilelike elements we have the peculiar situation that $E_n(t)$ *depends on* r. In the case of (C.1), the dependence enters implicitly through t_{nk}, while in the case of (C.4) it enters explicitly. This dependence occurs because of the internal steering, which always image-forms for a particular image point r. In the tilelike case let us temporarily write the voltage as $E_n(\mathbf{r}, t)$ (making the r-dependence explicit). Then in the discussion of image-forming, Equation (5.4) really means

$$\Pi(\mathbf{r}) = \sum_n w_n \sigma(\mathbf{r}, n) E_n \left[\mathbf{r}, \left(r + |\mathbf{r} - \mathbf{R}_n| \right) / c \right]. \tag{C.5}$$

On must therefore beware, when calculating $\Pi(\mathbf{r})$, that the E_n concerned has not only its normal dependence on r through its second argument, but also a dependence on r through its first argument.

Tilelike elements, like solid elements, are unshaded. Thus the image produced from an array of tilelike elements is not quite the same as if all the subelements had been converted into pointlike elements. For the pair of far-field conditions to hold, two conditions suffice. The first is the display equation that follows (3.7); the second is:

$$r / \cos^2(\mathbf{r}, \hat{z}) \gg (M'd)^2 / \lambda_c.$$

C.2. Case of Toneburst

For the case of a *toneburst* ($b = 0$), a closed expression can be obtained for the voltage. The derivation is omitted, but the result and some explanation of it are provided. Recall that the directivity function of a square array [Steinberg 1976, Ziomek 1985] is

$$D_t(u, v, f) = \frac{\sin\left(\pi f c^{-1} M' du \right)}{M' \sin\left(\pi f c^{-1} du \right)} \cdot \frac{\sin\left(\pi f c^{-1} M' dv \right)}{M' \sin\left(\pi f c^{-1} dv \right)}. \tag{C.6}$$

Subject to conditions similar to those for the solid element, the output voltage is found to be

$$E_n(t) = K_2 K_6 M'^2 \sum_j \frac{a_j}{r_j |\mathbf{r}_j - \mathbf{R}_n|} D_t \left(u_{jn} - u_n, v_{jn} - v_n, f_c \right) \xi \left[t - \tau(\mathbf{r}_j, n) \right]. \tag{C.7}$$

There are now four sine-of-angle variables: u_{jn} and v_{jn} defined by (3.9), together with

$$u_n = (x - X_n) / |\mathbf{r} - \mathbf{R}_n| \quad \text{and} \quad v_n = (y - Y_n) / |\mathbf{r} - \mathbf{R}_n|. \tag{C.8}$$

Because the tilelike element is steered, to obtain the voltage response of the element, one must replace u in the directivity (C.6) by the difference between the 'u' of the incoming plane wave and the 'u' of the image or steering point [Steinberg 1976, Ziomek 1985]. Hence one obtains the arguments in (C.7). The result (C.7) is the same as the result (Eqn 3.3) for pointlike elements except that K_3 is replaced by

$$K_6 M'^2 D_t \left(u_{jn} - u_n, v_{jn} - v_n, f_c \right).$$

Equation (C.7) is subject to a further pair of conditions, somewhat similar to (B.7) but omitted here.

Appendix D: Monte Carlo Calculations

For many purposes it is useful to calculate averages of beam patterns over a number of different specific arrays generated by the same generic array, in other words to do Monte Carlo simulations. The simulation automatically updates the seed vector once during each 'run' (each specific array). The program outputs for plotting both the image intensity based on an 'incoherent' average and one based on 'coherent' averages. To define these averages we anticipate Section 5.2, in which, at each image point, an in-phase image amplitude (Π or Π_p) and a quadrature amplitude Π_q are defined, along with a total image amplitude Π_t given by (5.21). Then the incoherent intensity is

$$\langle \Pi_t^2 \rangle = N_a^{-1} \sum_n \Pi_{tn}^2 . \tag{D.1}$$

The coherent average amplitudes are

$$\langle \Pi_p \rangle = N_a^{-1} \sum_n \Pi_{pn} \quad \text{and} \quad \langle \Pi_q \rangle = N_a^{-1} \sum_n \Pi_{qn} , \tag{D.2}$$

while the coherent intensity is the sum of the squares of the latter two quantities. In (D.1) and (D.2), n labels the specific array (n th array) and N_a is the number of specific arrays. On each left-hand side, the angular brackets stand for the sample average (average over arrays).

It is of interest to compare these simulation outputs with the values obtained theoretically for the point spread function in the cw case [Steinberg 1976, p. 143]. First, for a single point target, *the incoherent average intensity* (over many arrays) *at each angular displacement in the distant sidelobes, relative to the peak intensity, is* $1/N$, where N is the number of elements. In this result, pointlike elements and far-field conditions are assumed, as well as on-sphere conditions, i.e. image point having the same range as the target. Then the peak total intensity Π_t^2 (value at the target) is constant from run to run; the $1/N$ result is relative to this constant taken as unity. (The above restrictions on the $1/N$ result—especially the far-field condition—can be relaxed somewhat.)

For completeness, we give Steinberg's generalisation of the $1/N$ result to an arbitrary angular displacement and the corresponding result for the coherent amplitudes. For this purpose we must first introduce an array, 'the mean array', defined thus: at each location $\mathbf{R}$ in the array, replace the probability $p(\mathbf{R})$ that there is an element at $\mathbf{R}$ by the definite existence of an element at $\mathbf{R}$ but with strength (measured in terms of its voltage response) equal to $p(\mathbf{R})$. Let the subscript zero refer to the mean array. Then the incoherent mean intensity at a given image point (the mean again being over many arrays) is

$$\langle \Pi_t^2 \rangle = (1 - 1/N) \, \Pi_{t0}^2 + 1/N . \tag{D.3}$$

The coherent average amplitudes (over many arrays) are given by

$$\langle \Pi_p \rangle = \Pi_{p0} \quad \text{and} \quad \langle \Pi_q \rangle = \Pi_{q0} . \tag{D.4}$$

Comparing (D.3) with (D.4), we see that in the wings the coherent intensity (D.4) falls off more rapidly with angular displacement than does the incoherent intensity, since there is a $1/N$ 'floor' in (D.3) but not in (D.4).

Appendix E: 'Umbra' and Related Regions

We consider the case of a single point object j, and begin from the image-forming equation (5.5). An image point $\mathbf{r}$ will be said to lie in the *'umbra'* if, for every element n, the argument of ξ lies outside the interval $(-T/2, T/2)$; then none of the elements makes a contribution to Π, which is therefore zero throughout the region. Similarly, an image point lies in the *'fully-ensonified'* region if, for every element n, the argument of ξ lies inside the interval $(-T/2, T/2)$; then all of the elements contribute. The other image points lie in the *'penumbra'*; then some but not all of the elements contribute. The above statements refer to an uncorrelated pulse. For a correlated pulse, the interval becomes $(-T, T)$ and the corresponding results for the correlated case can be obtained by the replacement $T \rightarrow 2T$.

From this point on we approximate the path difference $\delta\Lambda$ (the difference between the path length from the projector to the element via the image point, and that via the point object) by the far-field expression. Consequently the true boundaries of the region will lie only approximately along the surfaces deduced. As discussed in Section 5.1.2, the approximation will be good at angular displacements from the point target no greater than those of the first few sidelobes. The approximation will also be good out to somewhat greater angular displacements if the target is close to the broadside axis; the approximation is increasingly in error further out.

The far-field approximation is based on the far-field expansion that is described below Equation (F.3) before assuming that $\mathbf{r}$ is parallel to $\mathbf{r}_1$. The approximation retains just one term beyond the leading term. Then it is found that the path difference $\delta\Lambda$, which is the numerator in the argument of ξ in the first line of (5.5), becomes

$$\delta\Lambda = 2(r - r_j) - (u - u_j)X_n - (v - v_j)Y_n \, , \tag{E.1}$$

where

$$u = \frac{x}{r}, \qquad u_j = \frac{x_j}{r_j}, \qquad v = \frac{y}{r} \quad \text{and} \quad v_j = \frac{y_j}{r_j} \tag{E.2}$$

are direction cosines. Alternatively, this far-field result may be obtained by the geometrical method used to obtain (B.2).

For a general direction specified by (u, v), in order to locate the boundary points r of the umbra, etc., one would now have to consider the four extreme points of the array, namely $(X_n, Y_n) = (\pm Md/2, \pm Md/2)$. For simplicity, we restrict attention to directions (u, v) with $v = v_j$. The points (r, u, v) satisfying this lie on a cone through the point object. Near the object the cone approximates to a plane; loosely it may be called the xz plane. Then in (E.1) the Y_n coordinate becomes irrelevant. Substituting in (E.1) (besides $v = v_j$)

$$X_n = \pm Md/2 \quad \text{and} \quad \delta\Lambda = \pm cT/2,$$

we obtain all the boundaries as

$$2\left(r - r_j\right) = \pm cT/2 \pm \left(u - u_j\right)Md/2 , \tag{E.3}$$

where all combinations of + and - are allowed. (Replace T by $2T$ for a correlated pulse.) The specific boundaries are shown in Figure 24, where it is seen that some parts of the 'boundaries' given by (E.3) are actually not boundaries between the above regions but lie within the penumbra zone.
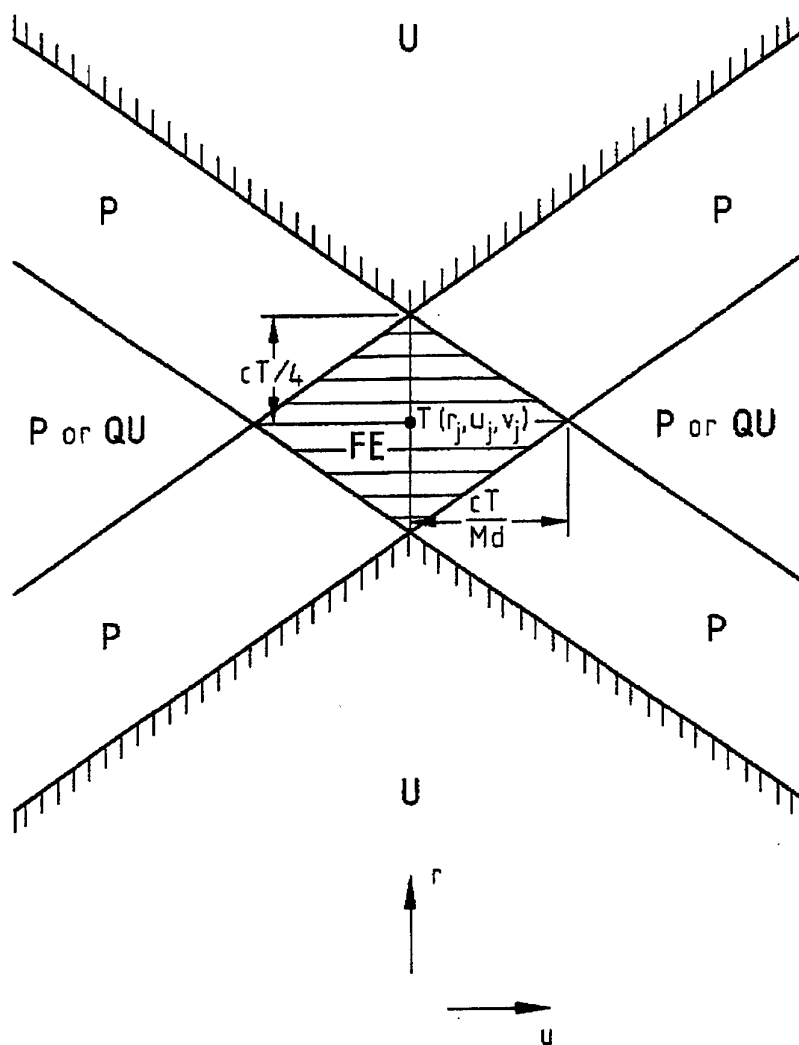
Figure 24: *Showing regions in the image of a point target located at T. Image points have coordinates $\left(r, u, v\right)$; the 'plane' $v = constant$ is shown. FE=fully-ensonified region, U=umbra or full-shadow region, P=penumbra, QU=quasi-umbra.*

Actually, the regions marked 'P or QU' in Figure 24 can be other than an ordinary penumbra. For these image points, all the elements that contribute have x coordinates that are interior to the region $\left(- Md/2, Md/2\right)$. Now consider an uncorrelated waveform (2.1) in which (2.12) holds, i.e. the sine passes through an even number of half-cycles. Then the contributions of the various elements are values of a complex number whose phase passes through a whole number of cycles as x passes through the 'allowed' values. For practically all arrays these contributions will come close to cancelling each other out. Then the image amplitude is quite small, and the region may be called a *quasi-umbra* region.

Appendix F: Image Amplitudes on the Radial Line through the Target

The purpose of this appendix is to derive, under certain conditions, expressions for the image amplitude along a radial line through the point target. The extension from uncorrelated to correlated signals is given in Section 5.1.3.2.

F.1. Pointlike Elements

Consider first pointlike elements. Substitution of (3.3) into the image-forming Equation (5.4) yields a certain equation for $\Pi(\mathbf{r})$, the in-phase part of the image amplitude. In the case of a single target ($j = 1$), that equation becomes

$$\Pi(\mathbf{r}) = K_3 \sum_n w_n \sigma(\mathbf{r}, n) \frac{a_1}{r_1 |\mathbf{r}_1 - \mathbf{R}_n|} \xi\left[\tau(\mathbf{r}, n) - \tau(\mathbf{r}_1, n)\right]. \tag{F.1}$$

In this appendix all the normalisation conditions of Section 4.3.1 will be imposed except (4.4); here condition (4.3) has been used. Now consider both $\mathbf{r}_1$ and $\mathbf{r}$ to lie in the *relaxed* far field (Eqn 4.5) of the array:

$$r_1 \gg Md, \qquad r \gg Md, \tag{F.2}$$

where Md is the side of the array. Then in the denominator of (F.1), $\mathbf{R}_n$ may be dropped. Similarly we may drop $\mathbf{R}_n$ in σ (see Eqns 5.23 and 5.24); in other words, in the calculation of σ, the nth element may be assumed to be relocated at the origin. Then (F.1) becomes

$$\Pi(\mathbf{r}) = K_3 \sum_n w_n \sigma(\mathbf{r}, 0) \frac{a_1}{r_1^2} \xi\left[\tau(\mathbf{r}, n) - \tau(\mathbf{r}_1, n)\right]. \tag{F.3}$$

To proceed, it is necessary to expand the square-root term in $\tau(\mathbf{r}, n)$, given by (5.3), in decreasing powers of the range r: the *far-field expansion*. Next, this expansion, with *two* terms retained beyond the leading term (the 'second-order approximation'), is applied to both τ s in (F.3). Then the fact that $\mathbf{r}$ and $\mathbf{r}_1$ are parallel enables us to simplify the argument of ξ in (F.3) to

$$\tau(\mathbf{r}, n) - \tau(\mathbf{r}_1, n) = \left\{ 2(r - r_1) + \tfrac{1}{2}\left(\frac{1}{r} - \frac{1}{r_1}\right)\left[R_n^2 - \left(\hat{r}_1 \cdot \mathbf{R}_n\right)^2\right] + \cdots \right\} / c; \tag{F.4}$$

note that the term that is first-order in $\mathbf{R}_n$ vanishes. In the curly brackets, two terms have been written. The second of these, T_2, is now dropped. The condition for the validity of this is that T_2 is small compared to λ_c, that is

$$\tfrac{1}{2} \frac{|r - r_1|}{r r_1}\left(\frac{Md}{\sqrt{2}}\right)^2 \ll \lambda_c, \tag{F.5}$$

93

where Md is the side of the array. In the summation over n in (F.3), all factors but w_n are now independent of n, so that the normalisation (4.8) may be invoked to yield the pointlike result (5.10).

F.2.   Solid Elements

The calculation for solid elements proceeds similarly to that for pointlike elements.[42] Equations (3.5) and (2.3) are substituted into (5.4); the right-hand side of (2.3) is to be evaluated at position $\mathbf{R}$ (physical spreading). Note that the origin of $\mathbf{R}$ is at the projector. For a single target ($j = 1$), we obtain exactly

$$\Pi(\mathbf{r}) = K_s \sum_n w_n \sigma(\mathbf{r}, n) \int_{S_n} \frac{a_1}{r_1 |\mathbf{r}_1 - \mathbf{R}|} \xi\left[\left(r + |\mathbf{r} - \mathbf{R}_n| - r_1 - |\mathbf{r}_1 - \mathbf{R}|\right)/c\right] dS . \tag{F.6}$$

Here $\mathbf{R}$ and $\mathbf{R}_n$ each appear in the argument of ξ (rather than $\mathbf{R}$ twice), because the element is unsteered. Next, the relaxed far-field condition (F.2) is invoked as for pointlike elements.

We now require that the target is in the far field of the *element*:

$$r_1 \gg (M'd)^2 / \lambda_c , \tag{F.7}$$

and let

$$\mathbf{R}' = \mathbf{R} - \mathbf{R}_n . \tag{F.8}$$

Using the binomial expansion in the usual way, we may write the last term in the argument of ξ as

$$|\mathbf{r}_1 - \mathbf{R}| = |\mathbf{r}_1 - \mathbf{R}_n - \mathbf{R}'| = |\mathbf{r}_1 - \mathbf{R}_n| - R' \cos \alpha_1 ,$$

where

$$\alpha_1 = (\mathbf{r}_1 - \mathbf{R}_n, \mathbf{R}') \tag{F.9}$$

where the parenthesis denotes the angle between the two vectors. Next, under the condition (F.5), a combination of four terms in the argument of ξ can be simplified as before. We then have

$$\Pi(\mathbf{r}) = K_s \sum_n w_n a_1 \frac{\sigma(\mathbf{r}, 0)}{r_1^2} \int_{S_n'} \xi\left\{\left[2(r - r_1) + \frac{\mathbf{R}' \cdot (\mathbf{r}_1 - \mathbf{R}_n)}{|\mathbf{r}_1 - \mathbf{R}_n|}\right]/c\right\} dS' \tag{F.10}$$

where the variable of integration has been changed from $\mathbf{R}$ to $\mathbf{R}'$.

We are interested in the conditions under which this expression reduces to an expression similar to those for pointlike elements. We require that the two subterms (from $\mathbf{r}_1$ and $-\mathbf{R}_n$ respectively) in the last term of the argument of ξ each be small compared to λ_c. That is,

$$|\mathbf{R}' \cdot \hat{r}_1| \ll \lambda_c , \qquad |\mathbf{R}' \cdot \mathbf{R}_n| / r_1 \ll \lambda_c .$$

---

[42] We avoid using the formula 3.11 because it holds only for a toneburst. The case $b \neq 0$ is relevant, because later we treat correlated signals by replacing ξ with Y.

These conditions reduce respectively to the sufficient conditions

$$\left|\left(\mathbf{r}_1, \hat{z}\right)\right| \ll \frac{\lambda_c}{M'd/\sqrt{2}}, \qquad r_1 \gg \frac{(Md)(M'd)}{2\lambda_c}. \qquad (F.11)$$

Of these, the first is a condition that the target is sufficiently near broadside. The second is a far-field condition, and requires that the target's range exceed a distance that is the geometric mean of the far-field transition distance of the array and the far-field transition distance of an element. Actually, from among the conditions on the solid element, condition (F.7) may be dropped, since the second condition in (F.11) already implies it.

Finally, under the above conditions, the integral over $\mathbf{R}'$ becomes a multiplication by $(M'd)^2$, which is the area of an element. Also invoking (4.8), we get (5.11) as the 'radial' result for solid elements.

F.3. Tilelike Elements

The calculation for tilelike elements parallels that for solid elements and will be omitted. The result is identical to (5.10) but with K_3 replaced by $K_5(M'd)^2$. The only conditions required are the conditions on pointlike elements, together with the requirement that the target point lie in the far field on an element.

F.4. Conditions of Validity

We summarise here the conditions of validity of the radial-line expressions for $\Pi(\mathbf{r})$, beginning with the case of pointlike elements, Equation (5.10). First it is required that both the target point and the image point lie in the relaxed far field (Eqn 4.6) of the array: the requirement is (F.2). Second, r must not be too different from r_1: the requirement is (F.5). For solid elements (Eqn 5.11), a third and a fourth condition are required: these are given by (F.11).

Appendix G: Demodulation in the Program: Conditions on and Errors in the Approximation

This appendix discusses the conditions and errors associated with the replacement (5.22). We concentrate on pointlike elements; solid elements are covered by the remark that, for them, the conditions in Section 3.2.2 need to be satisfied as well as the conditions in the current appendix.

G.1. Calculation of True Quadrature Waveform

We have ascertained the error, by first calculating the true quadrature waveform $\xi_q(t)$. To reduce the number of parameters, the scaled time

$$\tau = f_c t \tag{G.1}$$

was used, along with the scaled chirp duration

$$D = f_c T . \tag{G.2}$$

Thus the Fourier transform of ξ was taken with respect to τ, not t. ξ_q was calculated via the Fourier domain using (5.18).

Actually the fast Fourier transform (FFT), rather than the ordinary FT, was used. Due to the cyclic nature of the FFT, it was necessary to avoid end effects by embedding the toneburst in a long interval of zero voltage (Bergland 1969 discusses this point among the pitfalls of the FFT). Thus in our application, the FFT is characterised by two 'processing parameters': m, the number of sample points in the total interval, and θ, the fraction of the total interval that is occupied by the toneburst. The mathematical package MATLAB was used.

G.2. Conditions and Errors: Effect of Nonsmooth End

In accordance with Section 2.3.2, we consider a waveform that is ramp-end. Given that $\xi(t)$ is a toneburst (or more generally, that condition G.4 is satisfied), we estimate the resulting errors, beginning with the error in $\xi_q(t)$.

The true quadrature amplitude $\xi_q(t)$, obtained as in Section G.1, and the program's approximation to it, given by (5.22), are shown in Figure 25(a) for a *typical* short toneburst with a ramp ending.
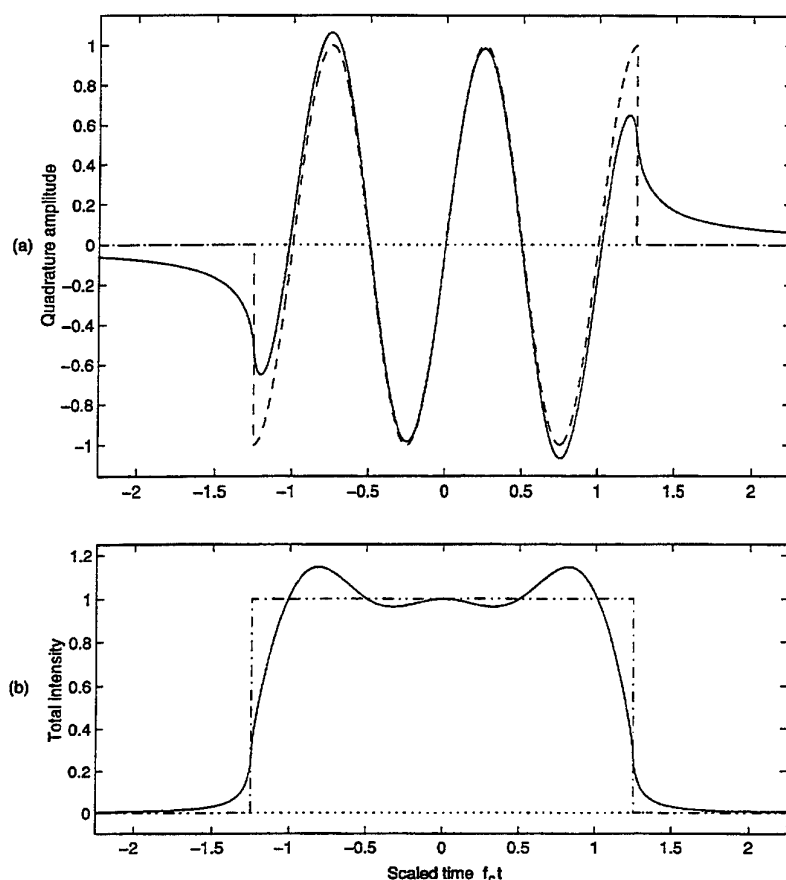
Figure 25: *Effect of the program's approximation on the 'total intensity' of the waveform and hence on the point spread function along a radial line through the target. The waveform is that of Figure 6(b), having a ramp ending. (a) shows the true quadrature amplitude $\xi_q(t)$ (continuous line) and the program's approximation to it (dashed). (b) shows the total intensity (continuous) and the approximation to it. The curve in (a) was obtained by a numerical Hilbert transform using the fast Fourier transform (FFT). To reduce end effects in the FFT to an imperceptible level, the pulse window was embedded in a large interval filled with zeros. Processing parameters: $m = 16\,384$, $\theta = 3/40$.*

While the approximation to $\xi_q(t)$ has a step-function end at $t = \pm T/2$, the true $\xi_q(t)$ falls fairly smoothly towards zero. The *considerable* error in the approximation that therefore occurs near $\pm T/2$ is confined to four intervals of $1/4$ cycle, one on each side of $t = \pm T/2$. For a few cycles beyond each of these four intervals, there is a *small* but not negligible error. The *error* in the quadrature amplitude is plotted in Figure 26. Also shown is a rough approximation to the error, which will be used in Section L.1.

The effect on the *image intensity* is most clear in the case when there is a single point target and the image point lies on a radial line through the target. For, as discussed in Section 5.1.3, the image amplitude function Π is then an accurate reflection of the waveform. Hence the total image intensity is proportional to the 'total intensity' of the waveform, defined as

$$\left[\xi_t(t)\right]^2 = \left[\xi_a(t)\right]^2 = \left[\xi(t)\right]^2 + \left[\xi_q(t)\right]^2.$$

This intensity $\xi_t^2(t)$ is shown in Figure 25(b) for the same waveform as in 25(a). It is seen that, as for the quadrature part of the waveform, the image intensity is in considerable relative error *only when* the range r is within 1/4 cycle, or $c/(8f_c)$ in distance terms, of the *cutoff points*. The figure also shows that, beyond the *outer* two of the four intervals, the error in the *intensity* is particularly small.

At any fixed, nonzero angular displacement from the target, the corresponding radial line passes through two 'umbra' regions, two 'penumbra' regions and either one or no 'fully ensonified' region (as discussed in Appendix E). We may expect the error to be small except near the umbra-penumbra boundaries.

Some further effects of the waveform's having a nonsmooth end are discussed in Section G.4.

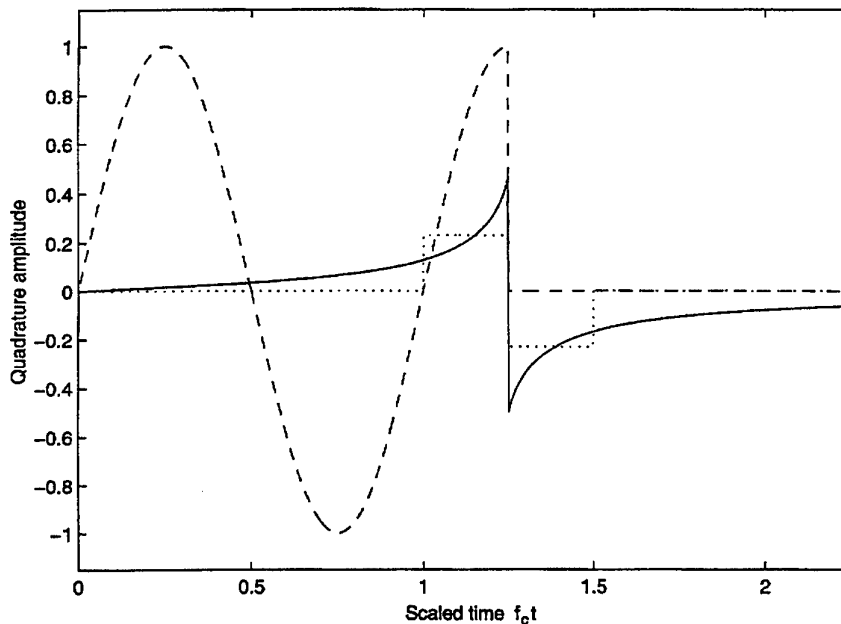*Figure 26: Error in the quadrature amplitude, due to the program, from the data displayed in Figure 25, the error being predicted amplitude minus true amplitude. Shown are the program's approximation (dashed), the error (continuous) and a rough approximation to the error (dotted). The latter approximation consists of two rectangle functions, each of height 0.23 and width 0.5; the value outside the rectangles is taken to be zero.*

G.3. Conditions and Errors: Short Toneburst

We now focus on the conditions and errors for a *short* toneburst—because shortness is needed in practice to achieve good range resolution. Let us compare (i) the size of one voxel in the range direction, namely $cT/2$, with (ii) the combined size of *two* of the four intervals in which there is considerable error (discussed in Section G.2). The size (ii) is chosen as the measure of error because the continuous interval over which $\xi_q(t)$ is given accurately is shrunk by the approximation by this amount. The 'continuous interval' from which shrinkage begins is equal to (i); thus the ratio of (ii) to (i) gives the relative shrinkage. The distance (ii) is twice one-quarter of a period, or in range terms, $c/2$ times this, namely $c/(4f_c)$. Thus the ratio, this time of (i) to (ii), is

$$\frac{\text{voxel size}}{\text{shrinkage}} = \frac{cT/2}{c/(4f_c)} = 2f_c T. \tag{G.3}$$

If this ratio is large, the shrinkage is relatively small and the approximation (5.22) may be said to be good overall.

For a numerical estimate, consider the case in which the range resolution is $\Delta_0 r = 1\,\text{mm}$ (the aim for the operational system) and the central frequency is 3.5 MHz. Then the pulse length is $T = 2\Delta_0 r/c = 1.33 \times 10^{-6}\,\text{s}$, and the ratio of sizes in (G.3) is 9.3. As this is quite large compared to one, the program's approximation is 'good overall'—though not 'very good.'

In practice the ratio (G.3) should be rounded up to 10 or down to 8 to satisfy Equation (2.12) or (2.13). This could be done, preserving the range resolution and pulse length, by altering the central frequency to 3.75 MHz or 3.0 MHz respectively. The pulse shapes for these two values (10 and 8) of the ratio are shown in Figure 6 (b and c respectively).

G.4. Waveforms with Nonsmooth Ends: Further Effects

This section describes additional effects arising from ramp-end waveforms and also step-function-end waveforms.

For a signal $\xi(t)$ with a step-function end (Section 2.3.1), the Hilbert transform, and hence the quadrature amplitude $\xi_q(t)$, unfortunately have a logarithmic *singularity* (behaviour as for $\ln|x|$ as $x \to 0$), lying precisely at each end of the pulse, just as the transform of the rectangle function itself is known to have. Figure 27(a) shows the quadrature amplitude for the step-function-end waveform displayed in Figure 6(d). The error in the program's approximation to $\xi_q(t)$, for the same waveform, is plotted in Figure 28.
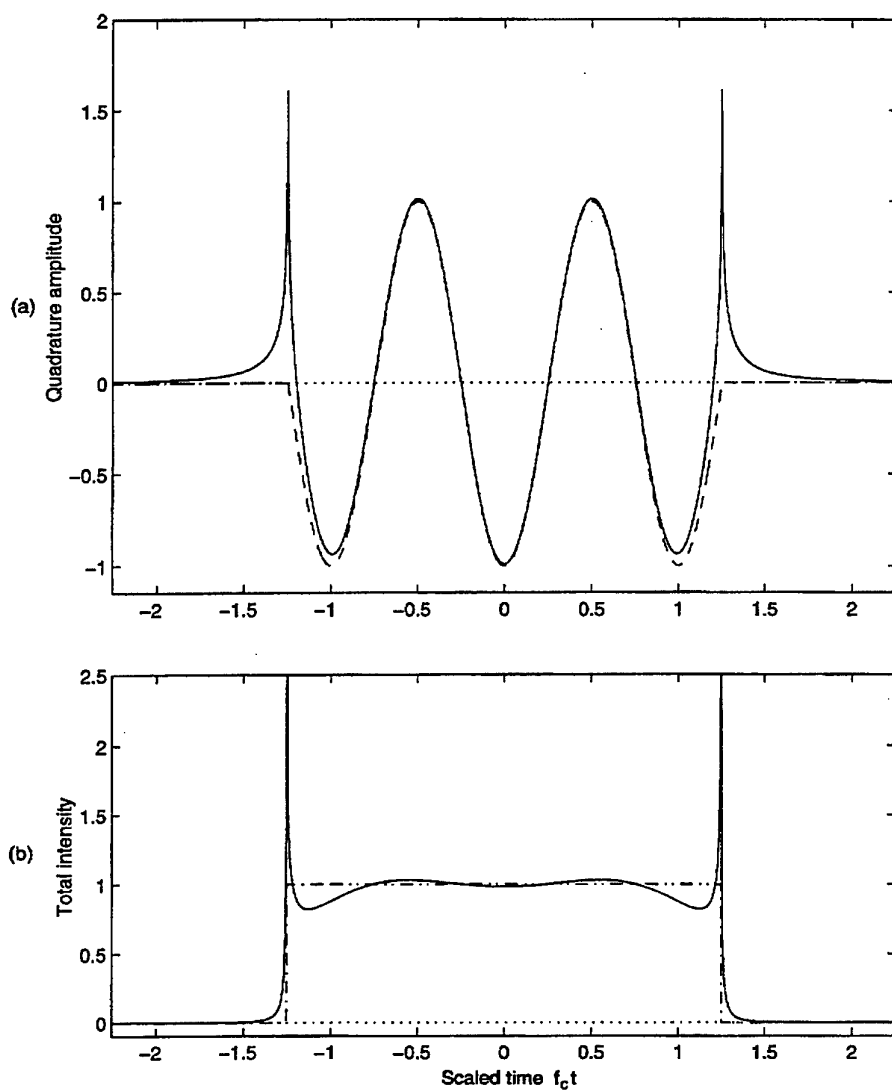
Figure 27: *As for Figure 25, but for the waveform of Figure 6(d), having a step-function ending. Note the logarithmic singularities, which however are each associated with a rather small area under the curve.* $m = 16\,384$, $\theta = 3/40$.
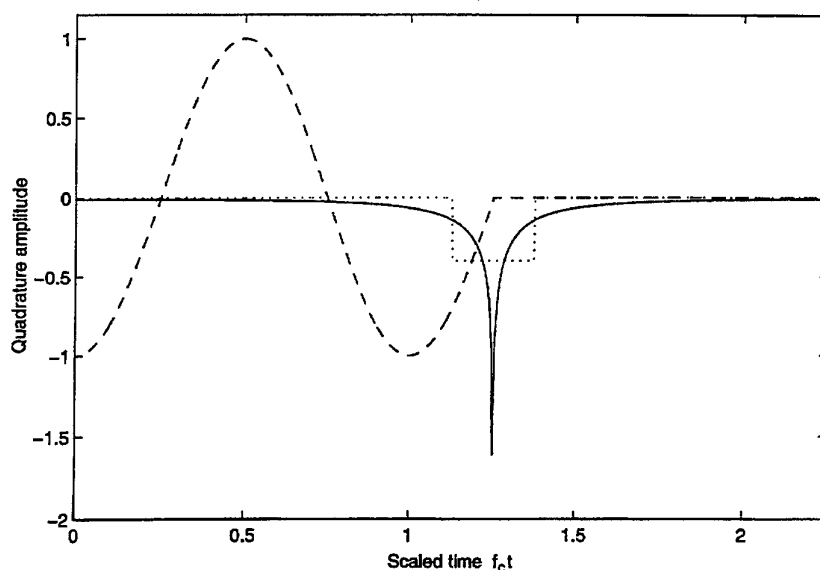
Figure 28: *Error in the quadrature amplitude, due to the program, from the step-function-end data shown in Figure 27. Shown are the program's approximation (dashed), the error (continuous) and a rough approximation to the error (dotted). The latter approximation consists of a single rectangle function of height 0.40 and width 0.25.*

For a ramp-end waveform, the $\ln|x|$ singularity is replaced by the more tame $x\ln|x|$ (as $x \to 0$) singularity. This 'tame' singularity can be seen at each end of the pulse in Figure 25(a).

Both the above singular behaviours are reflected closely in the behaviour of the image intensity along the radial line through the centre of the point spread function, through the range-time relationship discussed in Section 5.1.3.1. The respective total intensity functions are shown in Figures 27(b) and 25(b).

For the ramp-end waveform, any practical deleterious effect of the singularity is believed to be very small, and to be covered already by the stated errors in the program's approximation, given in Sections G.2 and G.3. This claim is borne out by results of D.E. Robinson [private communication], who found that the image from a ramp-end waveform and the image from the same waveform but in which the abrupt change in slope is smoothed out over (say) one-eighth of a cycle on either side of the abrupt change, are indistinguishable in practice.

Even step-function-end waveforms, with their logarithmic singularities, are not all that bothersome, given that the computed true image remains in fair agreement with the program's prediction along most of the 'ensonified' part of the central radial line (Section G.3).

G.5. Conditions and Errors: Other Results

A set of conditions of validity applying to more general results is included here, because it will be used in the later discussion on crosscorrelation.

Consider the more general waveform $\xi(t)$ of the form (2.10); the envelope may be smooth, but may be rectangular as a special case; and we may have $b \neq 0$. Consider the approximation that consists in putting the quadrature signal $\xi_q(t)$ equal to the same expression (2.10) but with the cosine replaced by sine — i.e. the analogue of (5.22). The conditions required for the validity of this replacement are twofold.[43] The first condition is that the relative change in frequency in one cycle be small:

$$|b| \ll f_c^2 \,. \tag{G.4}$$

Note that a toneburst has $b = 0$ and therefore satisfies (G.4) trivially. The second condition is that the pulse contains many cycles within its effective length:

$$f_c T' \gg 1 \,. \tag{G.5}$$

Here T' is some effective length of the pulse. T' can be conveniently chosen as $2\sqrt{3}$ times the standard deviation, so as to reduce to T in the case of a rectangular pulse. (Discussion elsewhere in the report suggests that when $f_c T' = 5$, the error is small though not negligible.)

---

[43] In the case of rectangle-type envelopes, there are also end effects, namely those discussed in Sections G.2 and G.4.

103

Appendix H: Relation to Ambiguity Function

The ambiguity function $\chi(\tau, v)$ of Rihaczek [1985, p. 119] is defined as

$$\chi(\tau, v) = \int_{-\infty}^{\infty} \mu(t)\, \mu^*(t - \tau)\, e^{j2\pi v t} dt, \tag{H.1}$$

where

$$\mu(t) = \psi(t)\, e^{-j2\pi f_0 t} \tag{H.2}$$

is called the *'complex envelope'*, the asterisk denotes complex conjugate, $\psi(t)$ is the analytic signal, and the carrier frequency f_0 [Rihaczek 1985, p. 14] is defined as the mean frequency of one side $(f > 0)$ of the energy density spectrum. The unnormalised autocorrelation function $Y_u(t)$ (numerator of Eqn 6.4) is not quite the result of putting $v = 0$ in the expression (H.1) for $\chi(\tau, v)$, for two reasons: first, because $Y_u(t)$ is the autocorrelation function of the in-phase, not the analytic, signal, and second, because what appears in (H.1) is not the analytic signal itself but the complex envelope. Nevertheless, a link can be established. Thus, letting $Y_{au}(t)$ be the unnormalised autocorrelation function of the *analytic* signal, defined by

$$Y_{au}(t) = \int_{-\infty}^{\infty} \xi_a^*(t')\, \xi_a(t + t')\, dt', \tag{H.3}$$

it readily follows that

$$Y_{au}(t) = e^{j2\pi f_0 t}\, \chi(t, 0). \tag{H.4}$$

Appendix I: The Quadrature Part of the Autocorrelation Function

As in Rihaczek [1985, p. 32] and Ziomek [1985], we make use of the *analytic signal* $\xi_a = \xi + j\xi_q$. With complex crosscorrelation defined by the display equation that precedes (6.2), consider the following quantity (much care being required with the signs):

$$\xi_a^* \otimes \xi_a = (\xi - j\xi_q) \otimes (\xi + j\xi_q)$$
$$= \xi \otimes \xi + j(\xi_q \otimes \xi) + j(\xi \otimes \xi_q) - \xi_q \otimes \xi_q.$$

But from (J.1), the Fourier transform of the left-hand side is

$$\mathcal{F}(\xi_a^* \otimes \xi_a) = \xi_a(-f)\xi_a(f).$$

On the right side of this last equation, the second factor is zero for all negative f (general property of analytic signals), while the first factor is zero for all positive f. The Fourier transform on the left is therefore identically zero. The expression on the left in the previous display equation is therefore *zero*. Consider now the last of the three expressions in that equation. Equating the real and imaginary parts to zero, we obtain the two rightmost equalities in (6.7).

The fact that the second and third expressions on the last line of (6.7) are equal to $K_4 Y_q$ follows immediately from Equation (J.3). [Note that in the present proofs, as in the proof of the evenness of $Y(t)$, the denominator K_4 can be treated as a constant; it depends on T but not on t.]

Appendix J: Alternative Expressions for the Quadrature Crosscorrelation Function

Consider two arbitrary *complex* functions f and g; for convenience in what follows, we write $f_1(\tau) = f^*(\tau)$. Recall the definition of crosscorrelation, given in the display equation preceding (6.2). Then, from the standard result for the Fourier transform of a *convolution* [Bateman and Yates 1988], it readily follows that the transform of a *crosscorrelation* is given by

$$\mathcal{F}\big[(f \otimes g)(\tau)\big] = F_1(-\phi)G(\phi), \tag{J.1}$$

where $G(\phi)$ and $F_1(\phi)$ respectively denote the Fourier transforms of $g(\tau)$ and $f_1(\tau)$. It can readily be shown that $F_1(-\phi) = F^*(\phi)$, so that we may rewrite (J.1) as

$$\mathcal{F}\big[(f \otimes g)(\tau)\big] = F^*(\phi)G(\phi). \tag{J.2}$$

In what follows, we consider only the special case where both f and g are *real* functions, so that Equation (5.18), involving the Fourier components of the Hilbert transform of h, holds both for f and for g. Then, using the above results, it is readily shown that

$$\mathcal{H}(f \otimes g) = -\hat{f} \otimes g = f \otimes \hat{g}. \tag{J.3}$$

From (J.3), the equality of the three expressions (Eqn 6.10, first line of 6.9, and second line of 6.9) for E_{rq_n} follows immediately.

Appendix K: Alternative Derivation of the Program's Expressions for the Autocorrelation Functions

Consider the true analytic signal $\xi + j\xi_q$, the exact autocorrelation function Y and its Hilbert transform Y_q. It is readily verified, from (6.7), that Y and Y_q are given *exactly* by a constant times the complex autocorrelation function:

$$Y + jY_q = \left[1/(2K_4)\right]\left(\xi + j\xi_q\right) \otimes \left(\xi + j\xi_q\right),\tag{K.1}$$

where complex correlation is defined by the display equation that precedes (6.2). Now suppose that (6.14), the approximation to $\xi_q(t)$, is substituted into (K.1) to give approximations to Y and Y_q. Then it turns out that the results (6.17) and (6.18) are obtained exactly; that is, *there is no need to make the further approximation of dropping the 'sum frequency' terms!* (This result has been obtained previously by Rihaczek 1985.)

This derivation *suggests* that the error in the program due to the dropping of the 'sum frequency' term in Section 6.3.1 is effectively zero — or at least much less than the value estimated in Section 6.5 and Appendix L. (There the estimation was based on the original derivation.) However computations given in that section and that appendix tell against the suggestion; we therefore put it aside.

Appendix L: Autocorrelation Functions: Details of Errors and Conditions

L.1. The Three Kinds of Error

The errors are of three kinds. The errors of the first two kinds are due to the replacement of the quadrature signal ξ_q by (6.14). From (6.4), it is seen that these two errors occur only in $Y_q(t)$, not in $Y(t)$. From (6.7), the combined (first plus second) error is

$$\delta Y_q(t) = K_4^{-1} \int_{-\infty}^{\infty} \xi(t') \delta\xi_q(t+t') dt'. \tag{L.1}$$

where $\delta\xi_q = \breve{\xi}_q - \xi_q$ is the error in ξ_q. For the present we restrict attention to ramp-end waveforms $\xi(t)$; then the error $\delta\xi_q$ is discussed in Section G.2 and plotted in Figure 26.

An argument based on the 'rough approximation' shown in that figure indicates that the error $\delta Y_q(t)$ has qualitatively different behaviour, depending on whether

$$0 \le t < T - 1/(4f_c) \quad \text{or} \quad t > T - 1/(4f_c). \tag{L.2}$$

The errors for the two respective intervals in (L.2) will be called the *errors of the first and second kinds* respectively. The first error covers almost all the interval of positive t up to the cutoff of $Y(t)$ at $t = T$ (and thus almost all the interval of range, centred on the target, from the one cutoff to the other), while the second error is appreciable only very near the cutoff.

The *error of the third kind* is due to the dropping of the 'sum frequency' term in Section 6.3.1. This third error will be discussed first. It will later be found that the error of the first kind makes no difference to the conclusions while the second error makes a difference only very near the cutoff of the Ys.

L.2. Error of the Third Kind

L.2.1 Theoretical Estimate of the Third Error

The error of the third kind is due to the dropping of the 'sum frequency' term in the derivation of (6.17) and (6.18). Thus $Y(t)$, for example, is in error by a definite integral of $\cos(B + A)$. We consider the case where Equations (2.1), (6.15), (6.16) and (6.25) are satisfied. An estimate of the error can be made by a slight modification of a principle used in optics [Jenkins and White 1950, p. 351; see also Ditchburn 1952, p. 200 and Neubauer 1986, Chapter 1]. The modified principle, derived in a separate section (Section L.6), is as follows. *The integral of a sinusoidal function of slowly varying amplitude and frequency, containing many cycles, with no point of near-stationary phase, is*

approximately one-half of the sum of the integral over the first half-cycle and the integral over the last half-cycle of the interval of integration.[44] For $Y(t)$, the sinusoidal function is $\cos(B + A)$. The result derived by applying this principle is that the rms error in each of Y and Y_q is

$$\Delta Y = \Delta Y_q = 1/(4\pi f_c T).$$ (L.3)

In the course of this derivation, a mean must be calculated to obtain the 'root-mean-square' value. At that point we note that the 'sinusoidal function' has a phase at the beginning of the first half-cycle and a phase at the end of the last half-cycle. We calculate the mean by simply averaging over an ensemble in which the difference in the two phases is distributed uniformly. From (L.3), the rms error in the total amplitude is

$$\Delta Y_t = \sqrt{2}/(4\pi f_c T),$$ (L.4)

i.e. Equation (6.28).

To discuss the *relative* error in amplitude, it is useful to concentrate on the range sidelobe *peaks*. Then in (6.19) we may replace the sine by unity; thus

$$Y_t = 1/(|b|Tt) = 1/(\pi Bt),$$ (L.5)

where Equation (6.22) has been used. Hence from (L.4) and (L.5), the relative error is

$$\Delta Y_t/Y_t = Bt/(2\sqrt{2}f_c T).$$ (L.6)

This relative error is certainly small for the 'close-in' sidelobes (i.e. $|t| \ll T$). Also, since $|t| \le T$, it is small everywhere if $B \ll f_c$ (i.e. small relative bandwidth). The formulae just derived (Eqns L.3 and L.4) will be tested in the remainder of this appendix.

L.2.2   Test of Formulae for the Third Error

For the present we assume that the errors of the first and second kinds are negligible; it will be seen from Sections L.3 and L.4 that this assumption justified. Thus the total actual errors in the Y s will be compared with the predicted formulae (L.3) and (L.4) for the third error. First, for the case $\beta = 1/3.5$, $f_c T = 40.25$ (Figures 12 to 14), the program's errors in the interval $0 \le t \le T$ were found to be essentially independent of t (Section 6.5.1). This finding is in agreement with the theoretical predictions (L.3). Second, the rms error was found to be (6.26); the prediction from (L.3) is

$$\Delta Y = \Delta Y_q = 0.0020,$$ (L.7)

in surprisingly good agreement.

Third, for the same waveform parameters, Figure 29 plots the total amplitude Y_t, the program's prediction $\breve{Y}_t$ and the error $\delta Y_t = \breve{Y}_t - Y_t$, plotted as 10 times the true value. The computed rms error $\Delta Y_t = 0.0032$ is equal to $\sqrt{2}$ times the actual ΔY of

---

[44] Neubauer unfortunately speaks of '*quarter*-wave zones.'

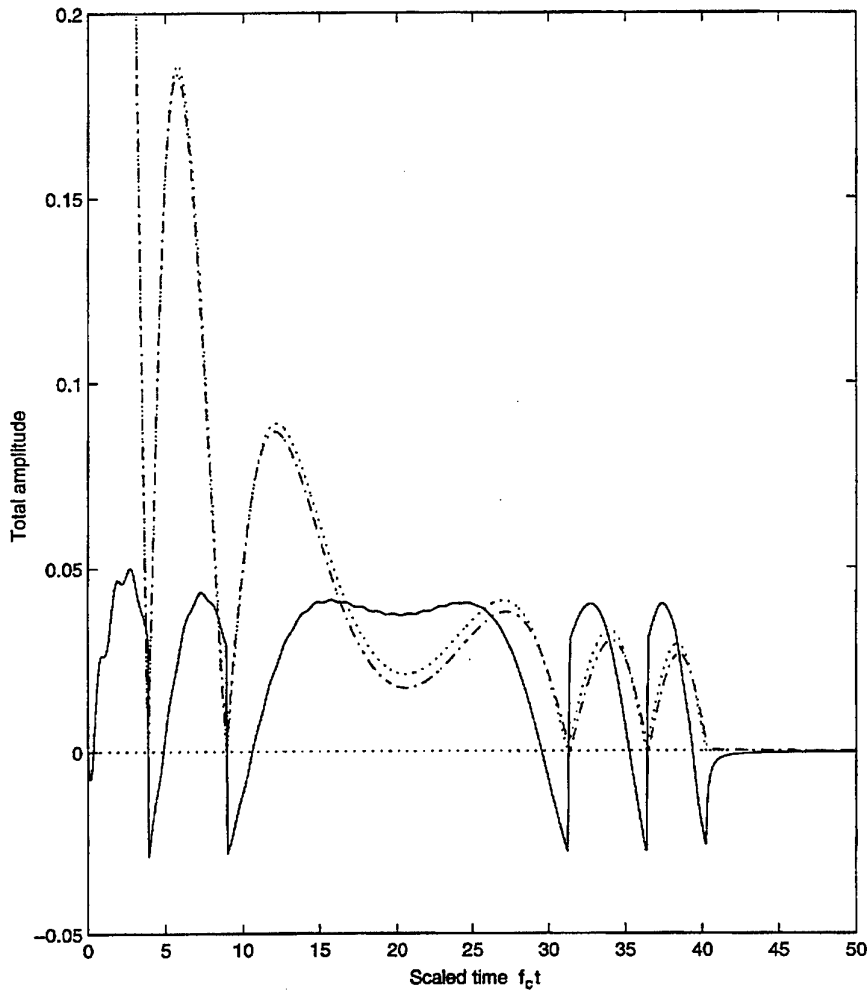(6.26) to within numerical errors; this confirms the concept of multiplying by $\sqrt{2}$ as in (L.4).

Figure 29: *Total amplitude Y_t of the autocorrelation function, for $\beta = 1/3.5$, $f_c T = 40.25$ (dash-dot). Also shown are the program's approximation (dotted) and 10 times the program's error (continuous). $m = 32\ 768$, $\theta = 1/14$.*
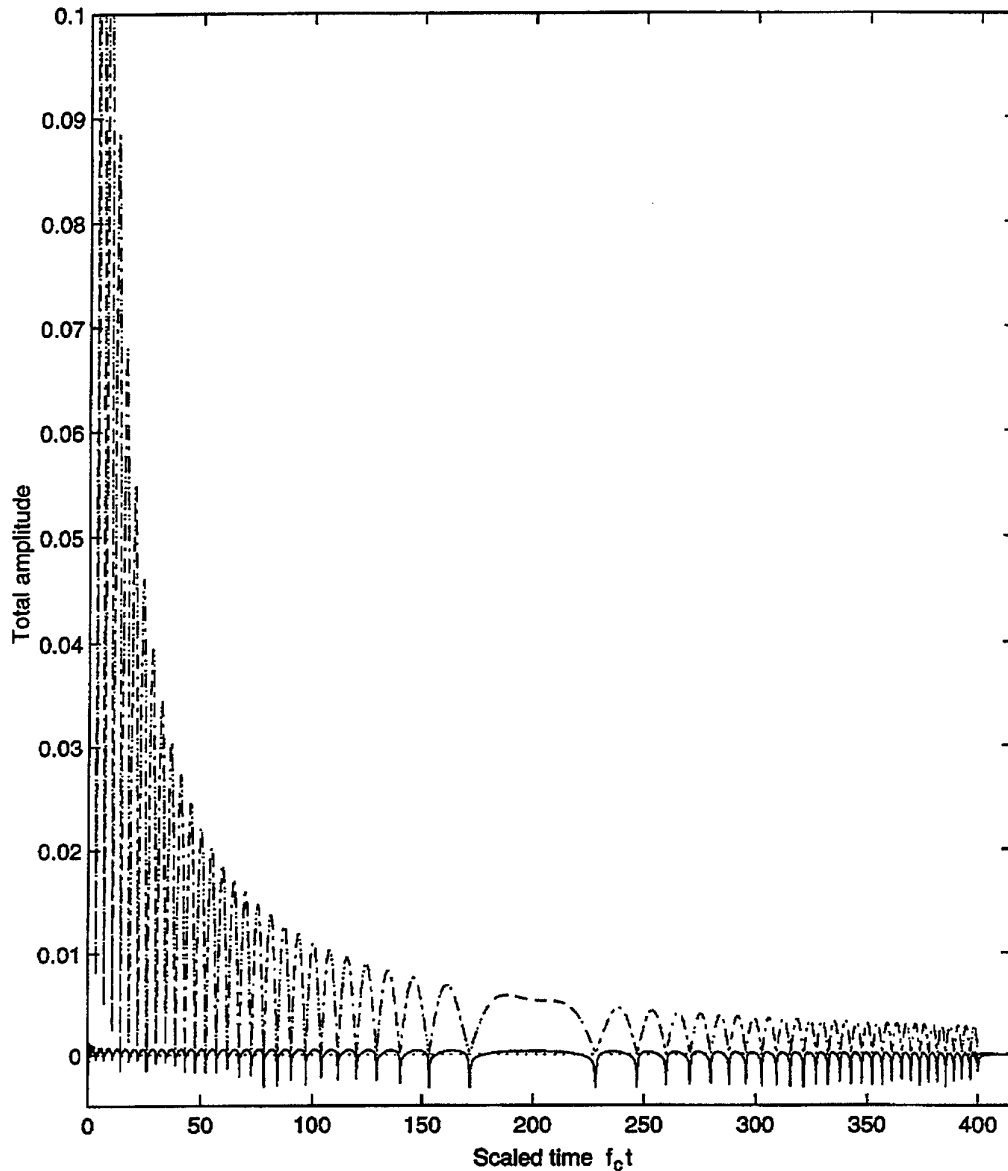
Figure 30: As for Figure 29, but for $\beta = 1/3.5$, $f_c T = 400$. $m = 65\,536$, $\theta = 1/14$.

We now move to a chirp containing many more cycles, as a more realistic example of the operating system. Figure 30 plots, for the parameters $\beta = 1/3.5$, $f_c T = 400$, the same quantities as in Figure 29. The true and predicted curves are *indistinguishable* on the scale of the graph. Actually the computed rms error in $0 \leq t \leq T$ is not zero but

$$\Delta Y_t = 5.4 \times 10^{-5}, \tag{L.8}$$

i.e. Equation (6.27). Gratifyingly, this is extremely small. The predicted error, from (L.4), is $\Delta Y_t = 2.8 \times 10^{-4}$. Thus the true error is one-fifth of the prediction; the reason for this better-than-expected result is not clear.

L.2.3   Relative Error in Last Sidelobe

It is of interest to look at the last range sidelobe before cutoff, as this is the lobe in which the program's relative error is expected to be worst ('worst sidelobe'). This expectation follows from the formula (L.6).

For the case $\beta = 1/3.5$, $f_cT = 40.25$ (medium-length chirp), an overall impression of the last sidelobe is given by a plot of the quadrature amplitude (Figure 31). For a measure of the error, note from Figure 29 that ΔY_t is about the same in the last lobe as elsewhere, so $\Delta Y_t = 0.0023$ from (6.26). From Figure 29, the true peak value is 0.026. So, relative to that peak, the rms error in amplitude is 9%.
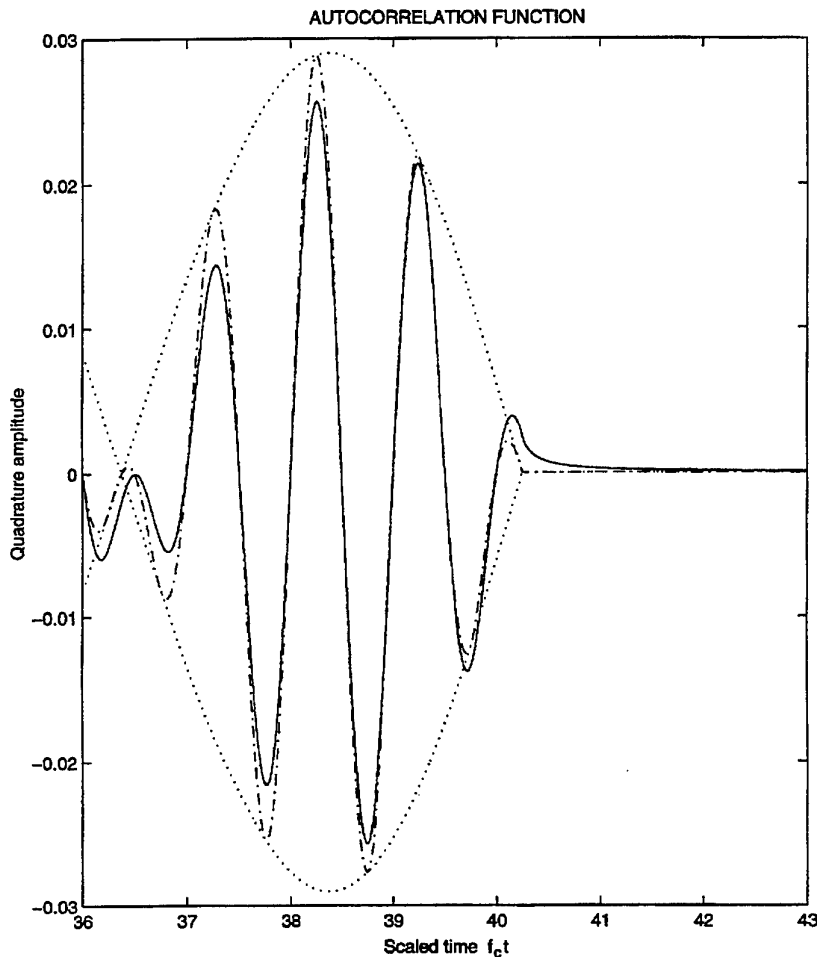
Figure 31: *For the last sidelobe before cutoff, the quadrature autocorrelation function* $Y_q(t)$ *(continuous) for* $\beta = 1/3.5$, $f_cT = 40.25$. *Also shown are the program's approximation (dash-dot) and the latter's envelope (dotted).* $m = 32\ 768$, $\theta = 1/14$.
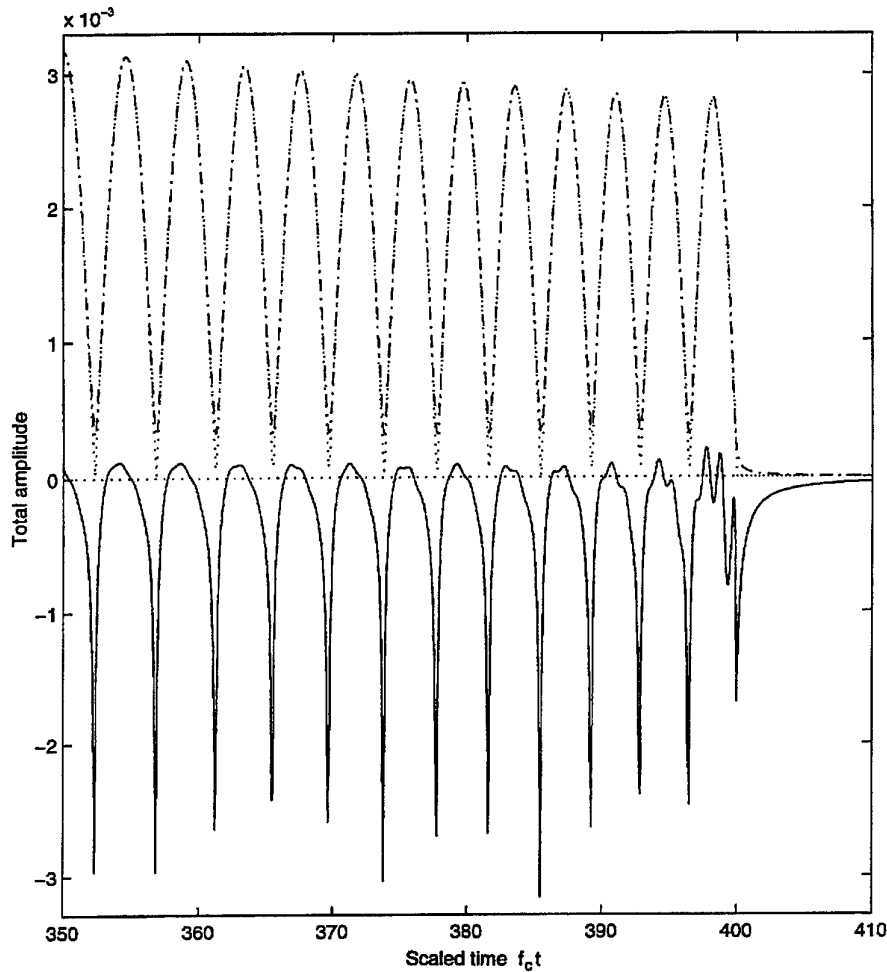
Figure 32: *For the last several sidelobes, the total amplitude Y_t of the autocorrelation function, for $\beta = 1/3.5$, $f_c T = 400$ (dash-dot). Also shown are the program's approximation (dotted) and 10 times the error in the latter (continuous). $m = 65\,536$, $\theta = 1/14$.*

For the case $\beta = 1/3.5$, $f_c T = 400$ (long chirp), the total amplitude of the last 12–13 sidelobes is shown in Figure 32. The computed rms error over just these sidelobes is found to be slightly different from the global average value (L.8): instead it is 6.6×10^{-5}. The relative error, expressed as in the previous paragraph, becomes 2.3% — considerably better than for the medium-length chirp.

118

L.2.4   The Third Error: Conclusion

The third error due to the program is, even at worst, fairly small, and is very small for most situations of interest. The theoretical formulae (L.3) and (L.4) for the error, while not correct for the whole range of situations, appear to give an upper limit on the error.

L.3.   Error of the First Kind

As mentioned in Section L.1, the error of the first kind occurs only in Y_q, not in Y. It should therefore show up as an excess of the rms error ΔY_q over ΔY. But the computed results in Section 6.5.1 consistently show no discernible excess; the two errors are equal. We conclude that the error of the first kind is negligible compared to that of the third kind.

L.4.   Error of the Second Kind

Again, this error occurs only for Y_q, not Y. The effect in the case $\beta = 1/3.5$, $f_c T = 40.25$ can be seen in Figure 31, in the last quarter-cycle or so before the cutoff, and also to the right of the cutoff. Roughly, the last peak of the carrier wave is stretched horizontally (without any change in the order of magnitude of the amplitude) to about 1/4 period into the 'umbra'. Beyond that point the amplitude falls off as $\left| t - t_{cutoff} \right|^{-1}$.

To check on the effect of a step-function end, the computations for Figure 31 were repeated with $f_c T = 40.0$, 40.5 and 40.75, to ensure that the cases covered a good range of strengths of the *step-function component* of the end of the chirp $\xi(t)$ (without calculating in advance what value of $f_c T$ would produce a pure ramp). The conclusions stated in the previous paragraph continued to hold for each of the three values of $f_c T$. Thus a step-function component does not lead to any marked increase in the second error.

The effect on the *total* amplitude may be seen in Figure 29, via the plot of the error (magnified).

We may therefore say that the error of the second kind is a rather small effect, the error in height being small compared to the height of the last sidelobe. That error is *very* small in the case of the long chirp $f_c T = 400$ (Figure 32).

L.5.   Conclusion

Thus the total error in the program's predictions may be identified with the error of the third kind (described by the conclusions of Section L.2.4), with the proviso that, in a

small region near the cutoff, small additional effects occur due to the error of the second kind. Regarding the size of the total error (third plus second), it is *rather small* even in the worst situations, and is *very small* for most situations of interest.

L.6. Optics-Type Integrals

This section gives the argument for the 'first and last half-cycles' result stated in Section L.2.1. The derivation is given even though it resembles a similar textbook proof [Jenkins and White 1950, p. 351], because there are two principal differences from that proof as follows. First, the conditions in the present result are more general than those in the optics result. Second, the optics result is always associated with the concept of Fresnel zones on the surface of a body, a concept not relevant to the present result.

Consider an integral of the form

$$I(t_1, t_2) = \int_{t_1}^{t_2} A(t)\, e^{j\theta(t)}\, dt,$$

where both the amplitude $A(t)$ and the angular frequency $d\theta/dt$ vary by only a small fraction of themselves in one cycle (i.e. change of 2π in θ). (This is the 'slowly varying' condition.) We first consider the case where $I(t_1, t)$ (which is the above integral but between the limits t_1 and t) converges as $t \to \infty$. (Here consider $A(t)$ and $\theta(t)$ to be continued smoothly into the region $t > t_2$, while still obeying the conditions of the stated result.) Take $e^{j\theta(t_1)}$, a constant, outside the integral sign, and break up the remaining integral into its real and imaginary parts. Then the real part of the integral resembles a cosine function (maximum value at $t - t_1 = 0$), the imaginary part a sine function. As discussed in Jenkins and White [1950, p. 353], or alternatively Ditchburn [1952, p. 201], the real part of $I(t_1, t)$ oscillates with t about a value equal to one-half the integral over the first half-cycle. The imaginary part oscillates about a value close to zero, which is again half the integral over the first half-cycle. In each case, as $t \to \infty$ convergence occurs to a value which is the said midpoint of the oscillation. Combining the real and imaginary parts, we obtain that $I(t_1, \infty)$ is half the contribution of the first half-cycle.

We may write

$$I(t_1, t_2) = I(t_1, \infty) - I(t_2, \infty).$$

The last term, equal to *minus* half the contribution of the half-cycle just beyond t_2, is also approximately *plus* half the contribution of the half-cycle just before t_2. Hence the result, for the convergent case.

When $I(t_1, \infty)$ is not convergent, one may replace the integrand by one that differs negligibly from $A(t)\, e^{j\theta(t)}$ in (t_1, t_2), is still smooth and obeys the conditions of the stated result in $t \geq t_2$, but eventually makes the new $I(t_1, \infty)$ convergent. The result then follows by the same argument as before.

Appendix M: Response of Extended Elements: Correlated Case

The response of solid elements is derived by first calculating the response of tilelike elements and then applying appropriate mathematical operations including the taking of a limit.

M.1. Tilelike Elements

From Equations (2.3), (C.1), (6.2) and (6.4), or alternatively from Section 6.7 and Equation (C.3), we obtain for the correlated output, without approximation,

$$E_{rn}(t) = K_2 K_6 \sum_j \sum_k \frac{a_j}{r_j |\mathbf{R}_n + \mathbf{R}_{nk} - \mathbf{r}_j|} Y \left[t - t_{nk} - \left(r_j + |\mathbf{R}_n + \mathbf{R}_{nk} - \mathbf{r}_j| \right) / c \right]. \tag{M.1}$$

(Figure 23 illustrates the geometry, provided that $\mathbf{R}'$ is replaced by $\mathbf{R}_{nk}$, the displacement of the kth subelement from the centre of the nth element.) When we make the far-field approximation as in Appendix B (the condition of validity being the display equation that follows Eqn 3.7), we obtain

$$E_{rn}(t) = \sum_j J_{jn} K_6 \sum_k Y \left[t - \tau(\mathbf{r}_j, n) + \left(X_{nk} \Delta u + Y_{nk} \Delta v \right) / c \right]. \tag{M.2}$$

Here

$$J_{jn} = K_2 \frac{a_j}{r_j |\mathbf{R}_n - \mathbf{r}_j|}, \tag{M.3}$$

$$\Delta u = u_{jn} - u_n, \qquad \Delta v = v_{jn} - v_n, \tag{M.4}$$

and the quantities on the right-hand sides of the last line are given by (3.9) and (C.8).

We proceed by transforming to the frequency domain, and then transforming back. This combined procedure enables the summation over subelements k to be performed and, as we shall see, also enables a rapidly converging series for $E_{rn}(t)$ to be developed. Using the Fourier transform defined by Equation (5.17) (but with t and f replacing τ and ϕ), we find that Equation (M.2) is transformed into

$$E_{rn}(f) = \sum_j J_{jn} K_6 \sum_k \exp \left\{ j2\pi f \left[-\tau(\mathbf{r}_j, n) + \left(X_{nk} \Delta u + Y_{nk} \Delta v \right) / c \right] \right\} Y(f) \tag{M.5}$$

The inverse transform yields

$$E_{rn}(t) = \sum_j J_{jn} K_6 \sum_k \int \exp \left\{ j2\pi f \left[t - \tau(\mathbf{r}_j, n) + \left(X_{nk} \Delta u + Y_{nk} \Delta v \right) / c \right] \right\} Y(f) df. \tag{M.6}$$

The (double) summation over $k = (k_1, k_2)$ can now be performed, since

$$X_{nk} = -(M' + 1)d/2 + k_1 d, \qquad Y_{nk} = -(M' + 1)d/2 + k_2 d. \tag{M.7}$$

where each of k_1 and k_2 runs from 1 to M', and the summation separates into the product of two (sums of) geometrical progressions. The result is

$$E_{rn}(t) = \sum_{j} J_{jn} K_6 \int \exp\left\{j2\pi f\left[t - \tau(\mathbf{r}_j, n)\right]\right\} M'^2 D_t(\Delta u, \Delta v, f) Y(f) df,\qquad\text{(M.8)}$$

where $D_t(u, v, f)$ is the tilelike-element directivity function given by (C.6) and rewritten at (M.12) below.

For enhanced understanding, we introduce

$$G(t) = K_6 M'^2 \int_{-\infty}^{\infty} D_t(\Delta u, \Delta v, f) Y(f) e^{j2\pi ft} df.\qquad\text{(M.9)}$$

$G(t)$ may be interpreted as the response of the element to the autocorrelated projected signal $Y(t)$ applied as the signal directly into the element, from the direction specified by (u_{jn}, v_{jn}), *without delay;* that is, applied without the delay of the round-trip time $\tau(\mathbf{r}_j, n)$. This is true because (M.8) may be written as

$$E_{rn}(t) = \sum_{j} J_{jn} G\left[t - \tau(\mathbf{r}_j, n)\right].\qquad\text{(M.10)}$$

Our task now is to perform the integration in (M.9) to obtain a result for $G(t)$ in terms of functions related to Y. For this purpose we develop a series for the directivity as follows. Let

$$a = d\,\Delta u/c,\qquad b = d\,\Delta v/c,\qquad\text{(M.11)}$$

where d is the spacing of the subelements; then from (C.6), the directivity function is

$$D_t(\Delta u, \Delta v, f) = \frac{\sin(\pi M'af)}{M'\sin(\pi af)} \cdot \frac{\sin(\pi M'bf)}{M'\sin(\pi bf)}.\qquad\text{(M.12)}$$

The numerator can be written in terms of complex exponentials and causes no trouble in the integration. To make the denominator amenable to integration over f, we express it as a power series in f, which we then divide into unity to obtain another power series. Thus the denominator becomes:

$$M'^2 (\pi af)(\pi bf)\left[1 - \pi^2 a^2 f^2/6 - \pi^2 b^2 f^2/6 + O(f^4)\right].$$

The series in square brackets can be divided into unity to yield a series

$$S = 1 + (\pi^2/6)(a^2 + b^2)f^2 + O(f^4),\qquad\text{(M.13)}$$

where S occurs in the directivity (M.12) according to

$$D_t(\Delta u, \Delta v, f) = \frac{\sin(\pi M'af)\sin(\pi M'bf)}{\pi^2 M'^2 abf^2} S.$$

Substitution of these results in Equation (M.9) yields

$$G(t) = \frac{K_6}{\pi^2 ab} \int_{-\infty}^{\infty} \sin(\pi M'af)\sin(\pi M'bf)\left[\frac{1}{f^2} + \frac{\pi^2}{6}(a^2 + b^2) + O(f^2)\right] Y(f)\, e^{j2\pi ft} df$$

$$\qquad\text{(M.14)}$$

We now write

$$G(t) = G_0(t) + G_1(t) + \ldots,\qquad\text{(M.15)}$$

where the term $G_m(t)$ is the contribution from the term in f^{2m} in S. It is readily shown that $G(t)$, $G_0(t)$ and $G_1(t)$ are all even in t.

The integrations in (M.14) can now be performed (see below). To express the results for the first two terms, we define

$$P(t) = \int_0^t \int_0^{t'} Y(t'')\, dt''\, dt' = \int_0^t (t - t')\, Y(t')\, dt' . \qquad \text{(M.16)}$$

Then $d^2 P/dt^2 = Y(t)$; also $P(t)$, like $Y(t)$, is an even function, and its derivative is odd. Then the first two terms of $G(t)$ are:

$$G_0(t) = \frac{K_6}{ab} \sum_\mu \sum_\nu \mu\nu P\left(t + \mu \frac{M'a}{2} + \nu \frac{M'b}{2} \right), \qquad \text{(M.17)}$$

$$G_1(t) = - \frac{K_6\left(a^2 + b^2\right)}{24ab} \sum_\mu \sum_\nu \mu\nu Y\left(t + \mu \frac{M'a}{2} + \nu \frac{M'b}{2} \right), \qquad \text{(M.18)}$$

where each of the summation indices μ and ν runs over the values -1 and 1 only.

The following notes will enable these results for the integrations in (M.14) to be checked. The term in $a^2 + b^2$ in (M.14) [namely $G_1(t)$] is readily integrated after expressing each sine in terms of complex exponentials. For each of the four resulting terms, the integral is an inverse Fourier transform but shifted in time; the result is (M.18). It is seen from (M.14) that the $1/f^2$ term [namely $G_0(t)$] is such that, if differentiated twice with respect to t, it would yield a constant times the term $G_1(t)$. Equation (M.17) satisfies this requirement with the appropriate multiplying constant. Actually this argument checks (M.17) only to within two constants of integration. However, an investigation shows these constants to be zero.

In summary, Equations (M.10) and (M.3) may be combined to yield

$$E_{rm}(t) = K_2 \sum_j \frac{a_j}{r_j |\mathbf{r}_j - \mathbf{R}_n|}\, G\left[t - \tau(\mathbf{r}_j, n) \right]. \qquad \text{(M.19)}$$

Then (M.19), (M.15), (M.17) and (M.18) together give the result for the in-phase correlated output voltage from a tilelike element, truncated to two terms. For the quadrature component, let us define $Y_q(f)$ as the Fourier transform of $Y_q(t)$ and $P_q(t)$ as the repeated integral of $Y_q(t)$ as in (M.16). $G_q(t)$ is defined by (M.9) but with $Y_q(f)$ on the right hand side. The correct quadrature results are obtained by adding subscripts q to the appropriate quantities in the equations that are given above for the in-phase case.

M.2. Solid Elements

The result for solid elements can be derived from that for tilelike elements. Five steps are needed. First, because the elements are unsteered, we make the replacements

$$\Delta u \to u_{jn}, \qquad \Delta v \to v_{jn} \ ,$$

as in Section 3.2 and Appendix A. Second, replace d throughout by δ —which eventually we shall make approach zero. Third, replace M' by $M'd/\delta$, so that the size of the element remains $M'd$ as $\delta \to 0$. Fourth, replace K_6 by $K_5 \delta^2$. And fifth, take the limit as $\delta \to 0$. We thus find, from Equations (M.15), (M.17) and (M.18), that

$$G(t) = G_0(t) = \frac{K_5 c^2}{u_{jn} v_{jn}} \sum_\mu \sum_\nu \mu \nu P \left(t + \mu \frac{M' du_{jn}}{2c} + \nu \frac{M' dv_{jn}}{2c} \right), \tag{M.20}$$

while $G_1(t)$ and all the higher terms *vanish*. As the limiting process has left $G_0(t)$ unaffected, it is clear that, in the expansion of $G(t)$ for the tilelike element (Eqns M.15, M.17 and M.18), the leading term represents a continuous (steered) tile, while the remaining terms are corrections due to the discreteness of the element structure. Finally, the limiting procedure leaves (M.19) unaffected, so that the Equations (M.20) and (M.19) give the in-phase result for solid elements.

When the quadrature component is calculated, $G_{q1}(t)$ and the higher terms vanish; and the final quadrature equations are (M.19) and (M.20), with a subscript q added on each side of each equation.

M.3. Extended Elements in the Program

The results of Sections M.1 and M.2 were coded in the program; at the same time the Ys were approximated as in Section 6.3.1. Prior to the forming of each image based on these formulae, tables of $P(t)$ and of the corresponding quadrature function are computed by numerical integration and stored. When needed, values of these functions are obtained by interpolation.

In the case of tilelike elements, the series for $G(t)$ and $G_q(t)$ (Equation M.15) are truncated after two terms. The condition for the validity of this (i.e. third term negligible) turns out to be

$$|\Delta u| << \left(\lambda_c / d \right), \tag{M.21}$$

where d is the subelement separation and Δu is defined by (M.4).

Appendix N: Quantisation

N.1. Quantisation in the Program

Recall that the program's image-former is to calculate the image amplitude at discrete positions $\mathbf{r}$. For each such $\mathbf{r}$, for each element n, there is a unique time $t = \tau(\mathbf{r}, n)$ at which the voltage and its quadrature part are to be evaluated (see Eqn 5.4). Only at these times is the quantisation operation actually performed. Let us denote by V any such voltage before quantisation, and by DV the quantised result (as if quantisation were performed continuously in time). Here D is the quantisation operator, defined as follows.

Let m (with $m \geq 1$) be the number of bits to which the voltages are quantised. The user selects either 'no quantisation' (in which case $DV = V$) or a particular value of m. In the latter case, the user must also specify the value of E_{max} (or equivalently, the value of f, see Eqn N.4). The allowed output values extend over some interval $(-E_{max}, E_{max})$ (Figure 15). This interval is subdivided into 2^m equal intervals; the midpoints of these intervals are at

$$D_p = -E_{max} + (2p - 1)\frac{E_{max}}{2^m} \qquad p = 1, 2, \ldots, 2^m . \tag{N.1}$$

These midpoints are the allowed output values of the quantisation operation. In the case of one-bit quantisation ($m = 1$), the value chosen for E_{max} is irrelevant, because a change in it changes all the image amplitudes by a constant factor only.

The algorithm for quantisation is as follows.

Step 1: Dithering.[45] A very small random number, having mean equal to zero, is added to V.

Step 2: The value of D_p that is closest to the dithered V is returned as DV. Then it can be shown that the quantised voltage is given by

$$DV = E_{max} \cdot \frac{\text{digsign}(x)}{2^{m-1}} \min\left[\text{int}\left(2^{m-1}|x|\right) + \tfrac{1}{2}, \; 2^{m-1} - \tfrac{1}{2}\right], \tag{N.2}$$

where

$$x = V/E_{max} + \varepsilon ,$$

and ε is a small random number, which POINTSPR takes to be uniformly distributed in the interval $(-0.5 \times 10^{-10}, 0.5 \times 10^{-10})$. In (N.2), $\text{int}(y)$ is the smallest integer less than or equal to y, while

$$\text{digsign}(x) = +1 \qquad x \geq 0$$
$$= -1 \qquad x < 0 .$$

---

[45] Dithering was found to be necessary because, without it, strictly zero values of V yield a DV always of the same sign—a result that would not hold in practice in the presence of the slightest noise.

In (N.2), the second argument of the minimum function allows for the fact that, no matter how large V may be, DV cannot rise above D_{2^m}.

N.2. The Optimal Value of E_{max}

For $m \geq 2$, it is desirable for the operator to have control of the value of E_{max} (the upper end-point of the uniformly quantised interval), for it can be used to *optimise the quantisation*.

Towards determining the optimum, consider the signals being received during some definite short time-interval within the received stream. Clearly the quantisation would be far from optimal if most of these voltages were small compared to E_{max}, as some quantisation levels would hardly be used. The quantisation would again be poor if the absolute value of the voltage exceeded E_{max} in most samples. Thus E_{max} *must be matched in some way to the typical or maximum size of the signals*. A suitable criterion for optimality — when the criterion can be applied — is that all the quantised values D_p be used equally often.[46]

Ideally a system — real or simulated — would continually modify the value used for E_{max} with time of reception in accordance with the signals received in a slightly earlier time interval — a kind of adaptive gain control. (Variation of average signal strength with time occurs, due in particular to the distribution of target strength over range and due to spherical spreading.) The program, however, uses the one value of E_{max} for all times. The question then arises: What is the best choice of this E_{max}? In the case of a single point target, or when all the point targets are at practically the same range, the appropriate course for the user is to optimise at the approximate time of reception that corresponds to that range. In the case of more general targets, the user could do different simulations (different runs of the program) to optimise at different ranges.

In summary so far, to select E_{max}, the user first selects the range (or equivalently, the time delay) of interest. Then, to the extent that the voltages are uniformly distributed over some interval, the quantisation is at its best when $\left(-E_{max}, E_{max}\right)$ is chosen to coincide with that interval.

We now derive a formula that estimates the optimal E_{max} — or at least an upper limit to it. (Note that the estimate makes use of knowledge of the total target; this knowledge is available only in the simulation!) This formula, given by the display equation below (N.4), should enable the optimal value of E_{max} to be obtained in the simulation with less effort than by beginning from an arbitrary initial choice.

First note that the voltage is subject to the bound

$$\left|E_n(t)\right| \leq \sum_j \left|a_j\right|/r_j^2 \tag{N.3}$$

---

[46] More generally (if the particular set of targets does not allow such equality), we may make the educated guess that the appropriate criterion is that the distribution of output voltages over the D_p is such as to maximise the entropy [Turner and Betts 1974, p. 21].

(where the inequality is approximate, not strict). In (N.3), the j are the targets and the constants K_1 to K_6 are chosen to have the values assigned in Section 4.3.1. To prove (N.3), consider the pointlike, uncorrelated case. Then in (3.3) we may take the absolute value of both sides and drop $\mathbf{R}_n$ from the denominator. $|\xi|$ may be replaced by its maximum value, unity, and thus (N.3) is obtained.[47,48]

Now let us write

$$E_{max} = f \sum_j |a_j|/r_j^2 ; \tag{N.4}$$

then a choice of f is tantamount to a choice of E_{max}. Consider the choice $f = 1$, yielding $E_{max} = E_{max1}$, where

$$E_{max1} = \sum_j |a_j|/r_j^2 .$$

Then the result (N.3) ensures that the voltages lie within $\left(-E_{max1}, E_{max1}\right)$ (approximately). The ideal E_{max} is therefore either close to, or less than, E_{max1}. We have thus found *an approximate upper limit on the ideal* E_{max}.

When there is just *one* target, a close look at the argument for (N.3) reveals that the voltage actually reaches (or gets very close to) the value E_{max1}; thus the voltage is distributed in what is *crudely* a uniform distribution, over $\left(-E_{max1}, E_{max1}\right)$. Hence, when there is one target, *the choice $f = 1$ should be near optimal*. On the other hand, when there are *many* targets, the spread (standard deviation) of the voltages about zero is usually much smaller than E_{max1}, because in general, in (3.3), the contributions to the voltage from the various targets add out of phase due to the differing path lengths. Then *some value of f considerably less than unity* should be optimal.

These guidelines serve to give the program user a suitable initial choice for E_{max}. The latter can then be tuned, so that by trial and error the user finds the value that gives the optimal image quality.

Note that, for a given scene (total target), there are two ways of achieving optimality: by varying E_{max} or by varying the power output of the transmitter. This is because, within limits, the system (that part of the entire system leading up to the digitiser) is linear. (For linearity, it is required that the ocean medium not be driven so hard that its behaviour becomes nonlinear, but also that the signals not become so low that they are comparable to, or smaller than, the noise.) Thus each scene is actually characterised by an optimal value of the ratio E_{max}/P_{tra}, rather than of E_{max} itself;

---

[47] For solid and tilelike elements the calculation is similar, based on Section 6.7 and the arguments given in Section F.2. For correlated signals, the result (N.3) follows readily using the arguments used to justify Equations (5.13) and (5.15).

[48] Incidentally, it is straightforward to strengthen the above argument leading to (N.3), to yield the same bound for the *analytic* signal:

$$|E_{an}(t)| \le \sum_j |a_j|/r_j^2 .$$

here P_{tra} is the transmitted pressure measured at some reference point. In the *operational system,* the operator would find the optimal value of the ratio by trial and error. Actually it is easier to design the system so that E_{max}, rather than P_{tra}, is kept fixed; the operator then determines the optimal value by varying P_{tra} [D.E. Robinson, private communication]. On the other hand, in the (present) *simulation,* P_{tra} is forced to be a constant and so the variation in the ratio must be achieved by varying E_{max} (or equivalently f).

DISTRIBUTION LIST

Underwater Acoustic Imaging: A Simulation Program and Related Theory

David G. Blair and Stuart D. Anstee

AUSTRALIA

DEFENCE ORGANISATION

Task Sponsor
    DGFD(Sea)
S&T Program
    Chief Defence Scientist
    FAS Science Policy   } shared copy
    AS Science Corporate Management
    Director General Science Policy Development
    Counsellor Defence Science, London (Doc Data Sheet)
    Counsellor Defence Science, Washington (Doc Data Sheet)
    Scientific Adviser to MRDC Thailand (Doc Data Sheet)
    Scientific Adviser Policy and Command
    Navy Scientific Adviser (3 copies)
    Scientific Adviser - Army (Doc Data Sheet and distribution list only)
    Air Force Scientific Adviser
    Director Trials
    Alan Burgess, LOD Salisbury
    Garry Newsam, SSD Salisbury

    Aeronautical and Maritime Research Laboratory
    Director

    Chief, Maritime Operations Division
    Research Leader Mine Warfare
    Dr B Ferguson, MOD Sydney
    Dr D Wyllie, MPD Melbourne
    Ranjit Thuraisingham, MOD Sydney
    John Shaw, MOD Sydney
    Kam Lo, MOD Sydney
    Gary Speechley, MOD Sydney
    Stuart Perry, MOD Sydney
    Doug Cato, MOD Sydney
    Mark Readhead, MOD Sydney
    Roger Neill, MPD Melbourne
    Ross Barrett, MOD Salisbury
    David Kershaw, MOD Salisbury
    David Blair (Author): 17 copies
    Stuart Anstee (Author) 2 copies

DSTO Library and Archives
    Library Fishermans Bend
    Library Maribyrnong
    Library Salisbury (2 copies)
    Australian Archives
    Library, MOD, Pyrmont (2 copies)
    Library, MOD, HMAS Stirling
    US Defense Technical Information Center, 2 copies
    UK Defence Research Information Centre, 2 copies
    Canada Defence Scientific Information Service, 1 copy
    NZ Defence Information Centre, 1 copy
    National Library of Australia, 1 copy
    The Director, Qinghua University, Beijing, 1 copy
    The Librarian, Beijing University of Aeronautics, 1 copy
    The Librarian, Institute of Mechanics, Chinese Academy of Sciences, 1 copy

Capability Systems Staff
    Director General Maritime Development
    Director General C3I Development (Doc Data Sheet only)
    Director General Aerospace Development (Doc Data Sheet only)

Navy
    SO (Science), Director of Naval Warfare, Maritime Headquarters Annex, Garden Island, NSW 2000. (Doc Data Sheet only)
    Deputy Director Mine Warfare Development
    SO (Science), CSOW, Garden Island, NSW 2000 (Doc Data Sheet only)
    Mine Warfare Systems Centre, PD
    MHC, PD
    COMAUSNAVMCDGRP

Army
    ABCA Standardisation Officer, Puckapunyal, (4 copies)
    SO (Science), DJFHQ(L), MILPO Enoggera, Queensland 4051 (Doc Data Sheet only)
    NAPOC QWG Engineer NBCD c/- DENGRS-A, HQ Engineer Centre Liverpool Military Area, NSW 2174 (Doc Data Sheet relating to NBCD matters only)

Intelligence Program
    DGSTA Defence Intelligence Organisation
    Manager, Information Centre, Defence Intelligence Organisation

Corporate Support Program
    Library Manager, DLS, Canberra

UNIVERSITIES AND COLLEGES
Australian Defence Force Academy
Library
Head of Aerospace and Mechanical Engineering
Serials Section (M list), Deakin University Library, Geelong, 3217
Hargrave Library, Monash University (Doc Data Sheet only)
Librarian, Flinders University

OTHER ORGANISATIONS
NASA (Canberra)
AusInfo
Mr Graeme Manzie, AMI Manager, Thomson Marconi Sonar, 274 Victoria Rd,
Rydalmere, NSW 2116 (4 copies)
Dr Mark Hedley, AMI Manager, TIP Division, CSIRO, PO Box 76, Epping, NSW
1710 (3 copies)
Dr Bob Harrison, Manager NDE, ANSTO, New Illawarra Rd, Lucas Heights,
NSW 2234 (2 copies)
Dr Ling Guan, Dept of Electrical Engineering, University of Sydney, NSW 2006
Dr ISF Jones, Ocean Technology Group, J05, University of Sydney, NSW 2006
Dr Andrew Madry, Ocean Technology Group, J05, University of Sydney, NSW
2006
Dr David Robinson, Pen-Y-Sarn, 383 Burraneer Rd, Coomba Park, NSW 2428
Mr JL Thompson, 52 Stokes Ave, Asquith, NSW 2077

OUTSIDE AUSTRALIA

ABSTRACTING AND INFORMATION ORGANISATIONS
Library, Chemical Abstracts Reference Service
Engineering Societies Library, US
Materials Information, Cambridge Scientific Abstracts, US
Documents Librarian, The Center for Research Libraries, US

INFORMATION EXCHANGE AGREEMENT PARTNERS
Acquisitions Unit, Science Reference and Information Service, UK
Library - Exchange Desk, National Institute of Standards and Technology, US

SPARES (5 copies)

Total number of copies: **105**

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA	1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)

2. TITLE Underwater Acoustic Imaging: A Simulation Program and Related Theory	3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U) Abstract (U)

4. AUTHOR(S) David G. Blair and Stuart D. Anstee	5. CORPORATE AUTHOR Aeronautical and Maritime Research Laboratory PO Box 4331 Melbourne Vic 3001

6a. DSTO NUMBER DSTO-TN-0274	6b. AR NUMBER AR-011-429	6c. TYPE OF REPORT Technical Note	7. DOCUMENT DATE April 2000

8. FILE NUMBER 510/207/0773	9. TASK NUMBER M40720	10. TASK SPONSOR DGFD(Sea)	11. NO. OF PAGES 128	12. NO. OF REFERENCES 41

13. URL on the World Wide Web http://www.dsto.defence.gov.au/corporate/reports/DSTO-TN-0274.pdf	14. RELEASE AUTHORITY Chief, Maritime Operations Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, SALISBURY, SA 5108

16. DELIBERATE ANNOUNCEMENT

No limitations

17. CASUAL ANNOUNCEMENT	Yes

18. DEFTEST DESCRIPTORS

Acoustic imaging, Beam forming, Sonar arrays, Signal processing

19. ABSTRACT
To aid in the design of a high-resolution sonar (range and transverse resolution approaching 1 mm), a program, POINTSPR, was produced. POINTSPR predicts the image obtained from an active acoustic imaging system with a spherical projector, the target being modelled by a small number of point objects. The waveforms available include chirps and short tonebursts. The receiving elements available include a point and a square piston. A wide variety of receiving array geometries are available, through features including the random placement of elements and the placing of multiple copies of a 'tile' of elements. Monte Carlo calculations can be run automatically. Sample graphs, which are 1-D or 2-D slices through the 3-dimensional (3-D) image, are given.
The theory of image-forming, dechirping and the quadrature part for a real sonar is presented, including 'delay-and-add' image-forming, known to apply even in the near-field, broadband, 3-D-imaging situation. The report also gives many formulae for image intensity within the simulation model. For a chirp of finite length, certain approximations, often applied, are studied numerically. Formulae, possibly new, are obtained for the response of square elements to a chirp.